

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia Eletrônica

Dispositivo de Monitoramento de Temperatura Via Wi-Fi Para Aplicações Laboratoriais

Autor: Filipe Alves de Sousa
Orientador: Prof. Dr. Roberto de Souza Baptista

Brasília, DF
2020



Filipe Alves de Sousa

Dispositivo de Monitoramento de Temperatura Via Wi-Fi Para Aplicações Laboratoriais

Monografia submetida ao curso de graduação em Engenharia Eletrônica da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof. Dr. Roberto de Souza Baptista

Brasília, DF

2020

Filipe Alves de Sousa

Dispositivo de Monitoramento de Temperatura Via Wi-Fi Para Aplicações
Laboratoriais/ Filipe Alves de Sousa. – Brasília, DF, 2020-
85 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Roberto de Souza Baptista

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2020.

1. Termopar. 2. Raspberry Pi. I. Prof. Dr. Roberto de Souza Baptista.
II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Dispositivo de
Monitoramento de Temperatura Via Wi-Fi Para Aplicações Laboratoriais

CDU 02:141:005.6

Filipe Alves de Sousa

Dispositivo de Monitoramento de Temperatura Via Wi-Fi Para Aplicações Laboratoriais

Monografia submetida ao curso de graduação em Engenharia Eletrônica da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica.

Trabalho aprovado. Brasília, DF, 17 de dezembro de 2020:

Prof. Dr. Roberto de Souza Baptista
Orientador

**Prof. Dra. Claudia Patricia Ochoa
Diaz**
Convidado 1

**Prof. Dra. Mariana Costa Bernardes
Matias**
Convidado 2

Brasília, DF
2020

Agradecimentos

A Deus, pela vida. Aos meus pais e irmãos, por me apoiarem em todos os momentos. A todos os professores que contribuíram com a construção do meu aprendizado durante a minha trajetória acadêmica. Aos meus colegas que ajudaram a descomplicar diversos problemas, de maneira descontraída, nos últimos anos.

Resumo

Este projeto propõe a implementação de um dispositivo capaz de medir uma larga faixa de temperatura, com base no Efeito Seebeck, permitindo o registro e monitoramento dos valores à distância, por meio do padrão de rede *Wi-Fi*. Com objetivo de facilitar análises específicas de variação de temperatura em experimentos laboratoriais, estabelecendo-se técnicas para automatizar o processo de modo que os resultados sejam mais rápidos e tão precisos quanto aqueles obtidos pelos métodos comumente utilizados. Foram realizados ensaios de medição de temperatura, onde obteve-se medidas por meio de um dispositivo comercial, estas foram anotadas manualmente e comparadas com medidas registradas automaticamente pelo dispositivo proposto, ao qual permite salvar localmente os dados e transmiti-los ao operador. O desempenho da aplicação destes métodos foi comparado, empregando análises estatísticas nos resultados obtidos. Os resultados relatados neste trabalho mostraram que é possível automatizar o processo de registro de medidas de temperatura. A automatização deste processo de registro mostrou uma interação satisfatória e passível de melhorias, sendo portanto, uma solução viável que pode ser replicada e otimizada para fins de aplicações em experimentos científicos.

Palavras-chaves: Efeito Seebeck. Temperatura. Raspberry Pi. Monitoramento. Instrumentação Eletrônica.

Abstract

This project proposes the implementation of a device capable of measuring a wide temperature range, based on the Seebeck Effect, allowing the recording and monitoring of values at a distance, through the Wi-Fi network standard. With the objective of specific analysis of temperature variation in laboratory experiments, defining techniques to automate the process so that the results are faster and as accurate as those obtained by the commonly used methods. Temperature measurement tests were carried out, where measurements were obtained by means of a commercial device, these were annotated manually and compared with measurements registered automatically by the proposed device, which allows the data to be saved locally and transmitted to the operator. The performance of the application of these methods was compared, using statistical analysis on the results obtained. The results reported in this work indifferent that it is possible to automate the process of recording temperature measurements. The automation of this registration process shows a satisfactory interaction that can be improved, being, therefore, a viable solution that can be replicated and optimized for the purposes of applications in scientific experiments.

Key-words: Seebeck effect. Temperature. Raspberry Pi. Monitoring. Electronic Instrumentation.

Lista de ilustrações

Figura 1 – Esquema representativo do circuito implementado por Seebeck. Fonte: O autor.	24
Figura 2 – Faixa de trabalho dos termopares tipo J, K e S. Tensão x Temperatura. Fonte: (ZUMBAHLEN, 2008).	26
Figura 3 – Esquema de um Termopar tipo K. Fonte: Texas Instruments, adaptado.	27
Figura 4 – Sensibilidade dos termopares tipo J, K e S. Coeficiente de Seebeck x Temperatura. Fonte: (ZUMBAHLEN, 2008).	27
Figura 5 – Plataformas: 1. <i>ESP32</i> , 2. <i>Raspberry Pi Zero Wh</i> e 3. <i>BeagleBone Black Wireless</i> . Vistas superiores. Fonte: (FILIPEFLOP, 2020), adaptado.	28
Figura 6 – Diagrama de blocos funcional do projeto - Fluxo de dados. Fonte: O autor.	33
Figura 7 – <i>Raspberry Pi Zero W</i> , vistas superior e inferior. Fonte: (FOUNDATION, 2017).	35
Figura 8 – Barra de pinos - <i>Raspberry Pi Zero W GPIO Header</i> . Fonte: (PI4J, 2012).	35
Figura 9 – Módulo medidor de temperatura da <i>Texas Instruments</i> , de um canal, juntamente com a sonda termopar tipo K. Fonte: (INSTRUMENTS, 2019a).	36
Figura 10 – Diagrama do circuito ADS1118 com canal duplo. Este circuito permite realizar registros por meio da configuração do multiplexador e de outros componentes embutidos no ADS1118. O multiplexador permite a leitura na entrada analógica e o envio de sinais, que podem ser amplificados, ao conversor analógico-digital. Estes sinais são filtrados e transmitidos à RPi. Fonte: (BECKMAN; CHIOYE, 2011).	37
Figura 11 – Diagrama da junção do termopar ADS1118. Fonte: (BECKMAN; CHIOYE, 2011).	38
Figura 12 – Diagrama de pinos do circuito do display LCD. Fonte: (INSTRUMENTS, 2013).	39
Figura 13 – Placa pré-furada, soquete e barras de pinos. Fonte: O autor.	40
Figura 14 – Interface de <i>hardware</i> – barramento de conexão SPI, onde os canais SCLK, CS, DIN e DOUT foram utilizados para fazer interface de RPi com o <i>display</i> . Fonte: BECKMAN; CHIOYE, 2011, adaptado.	41

Figura 15 – Ciclo de transmissão de dados de 16 <i>bits</i> . Quando os dados do registro de configuração não precisam ser lidos, os dados de conversão ADS1118 podem ser sincronizados em um curto ciclo de transmissão de dados, de 16 <i>bits</i> . Quando o canal SCLK é mantido baixo por 28 <i>ms</i> , a interface serial é redefinida e o próximo pulso SCLK inicia um novo ciclo de comunicação. Fonte: (INSTRUMENTS, 2019a).	42
Figura 16 – Conexão sendo realizada. Fonte: O autor.	43
Figura 17 – Protótipo em desenvolvimento. Dispositivo de monitoramento de temperatura via <i>Wi-Fi</i> . Fonte: O autor.	44
Figura 18 – Imagem da tela inicial do Sistema Operacional <i>Raspbian</i> . Fonte: O autor.	45
Figura 19 – Diagrama de blocos - Fluxo do <i>software</i> . Fonte: BECKMAN; CHIOYE, 2011.	47
Figura 20 – Exemplo do método ao converter de tensão (V) em temperatura (T), usando tabelas de consulta. Fonte: BECKMAN; CHIOYE, 2011.	48
Figura 21 – Dispositivo TM-902C com termopar tipo K. Instrumento medidor de temperatura usado durante o ensaio de medidas de temperaturas. Fonte: O autor.	49
Figura 22 – Fogareiro (WY-01B) utilizado durante o ensaio de levantamento de medidas pontuais, com a finalidade de transmitir calor por meio de contato, como uma fonte variável de temperatura. Fonte: O autor.	50
Figura 23 – Ensaio de levantamento de medidas pontuais. Medindo a temperatura em pontos fixos, na superfície aquecida do fogareiro elétrico, por meio dos dispositivos termopares, com a finalidade de comparar os valores obtidos e fazer ajustes contando com o auxílio de tabela de conversão de tensão/temperatura para termopar tipo K. Fonte: O autor.	50
Figura 24 – Diagrama geral da arquitetura do dispositivo medidor de temperatura. Fonte: O autor.	52
Figura 25 – Fluxograma simplificado de comunicação utilizando <i>socket</i> para criar uma interface <i>web</i> . Fonte: O autor.	52
Figura 26 – Trechos do código em <i>Java Script</i> , implementado utilizando-se, dentre outras ferramentas, <i>String()</i> e <i>split()</i> para dividir <i>strings</i> em matrizes de <i>substrings</i> e retornar novas matrizes. Fonte: Yousef, adaptado.	53
Figura 27 – Temperaturas medidas, a cada segundo, e enviadas ao usuário que acessa o sistema remotamente. Fonte: O autor.	54
Figura 28 – Resultado, em tempo real, da medida de temperatura e horário exibido. Fonte: O autor.	55

Figura 29 – Compartilhamento e registro de medidas de temperatura via comunicação sem fio. Por meio da conexão <i>Wi-Fi</i> , as informações podem ser enviadas utilizando-se a arquitetura cliente-servidor a qual foi implementada para facilitar o acesso e salvamento dos dados, utilizando um navegador <i>web</i> . Fonte: O autor.	55
Figura 30 – Escolhendo a função <i>log</i> , determinando um intervalo de medidas, um nome para o arquivo.csv e clicando em iniciar, pode-se registrar e salvar os valores lidos durante o ensaio de medidas de temperatura. Fonte: O autor.	56
Figura 31 – Medidas sendo registradas em memória externa, por meio da função " <i>log</i> ". Acompanhamento do processo utilizando o terminal. Fonte: O autor.	56
Figura 32 – Medidas sendo realizadas com o ADS1118 e salvas durante o ensaio de medição de temperatura. O período estabelecido altera a frequência de envio de dados de temperatura, isso implica diretamente no espaço de armazenamento destes em um determinado intervalo de tempo. Definindo 25 segundos como o período entre as medidas realizadas, os valores foram registrados e salvos por meio da função " <i>log</i> ". Fonte: O autor.	57
Figura 33 – Medidas realizadas utilizando o TM-902C e registradas de forma manual, para cada valor de temperatura de referência, durante o ensaio. Grupo 1. Fonte: O autor.	58
Figura 34 – Medidas obtidas para cada valor de temperatura de referência, registradas automaticamente durante o ensaio, por meio do ADS1118 em conjunto com o sistema implementado. Grupo 2. Fonte: O autor. . . .	59
Figura 35 – Ilustração do Efeito Termoelétrico ou Efeito Seebeck. Fonte: BECKMAN; CHIOYE, 2011.	60
Figura 36 – Gráfico de dispersão com barras de erro das médias dos valores medidos entre 25 °C e 505 °C, considerando o intervalo de confiança de 95%. - Via ADS1118 (Grupo 2) – IBM SPSS. Fonte: O autor.	62
Figura 37 – Gráfico de dispersão com barras de erro das médias dos valores medidos entre 25 °C e 505 °C, considerando o intervalo de confiança de 95%. - Via TM-902C (Grupo 1) – IBM SPSS. Fonte: O autor.	63
Figura 38 – Banco de dados com medidas repetidas. Onde TM-902C e ADS1118 são os dispositivos comparados e M[t1] refere-se as medidas, realizadas durante o ensaio, no intervalo de tempo 1, M[t2] no intervalo de tempo 2 e M[t3] durante o intervalo de tempo 3. Fonte: O autor.	64

Figura 39 – Teste de Normalidade <i>Shapiro-Wilk</i> , considerando o nível de significância de $p=0,5$ e o coeficiente de confiança de 95,5%. Observa-se que $\text{Sig.} > 0,05$. Fonte: O autor.	65
Figura 40 – Teste de Normalidade <i>Shapiro-Wilk</i> após o tratamento de dados com função Logarítmica, considerando o nível de significância de $p=0,5$ e o coeficiente de confiança de 95,5%. Observa-se que $\text{Sig.} < 0,05$. Fonte: O autor.	65
Figura 41 – Teste de Esfericidade considerando o nível de significância de $p=0,5$ e o coeficiente de confiança de 95,5%. Observa-se que $\text{Sig.} > 5\%$. Fonte: O autor.	66
Figura 42 – Teste <i>Pairwise Comparisons</i> . IBM SPSS. Nível de significância de $p=0,5$ e coeficiente de confiança de 95,5%. Fonte: O autor.	67
Figura 43 – Gráfico de medidas do Grupo 1 entre 25 °C e 505 °C - Post Hoc - SPSS. Fonte: O autor.	68
Figura 44 – Gráfico de medidas do Grupo 2 entre 25 °C e 505 °C - Post Hoc - SPSS. Fonte: O autor.	68
Figura 45 – Gráfico de comparação das médias dos Grupos 1 e 2 - <i>Post Hoc Sidak</i> - SPSS. Fonte: O autor.	69
Figura 46 – Planilha de análise de dados - Medidas de temperatura obtidas durante o ensaio com os dois tipos de medidores - Diferenças e médias obtidas para encontrar a média das diferenças, o desvio padrão e os limites inferior e superior do gráfico de <i>Bland-Altman</i> . Fonte: O autor.	70
Figura 47 – Tabela com o resultado do teste <i>T</i> . Nível de significância de $p=0,5$ e coeficiente de confiança de 95,5%. Fonte: O autor.	70
Figura 48 – Gráfico de diferença das médias - <i>Bland-Altman</i> - SPSS. Fonte: O autor.	71
Figura 49 – Tabela com o resultado do teste de regressão linear, para verificar se existe um viés nos dados analisados - SPSS. Nível de significância de $p=0,5$ e coeficiente de confiança de 95,5%. Fonte: O autor.	72
Figura 50 – Tabela de configuração do ADS1118. Fonte: (BECKMAN; CHIOYE, 2011).	79
Figura 51 – Código em C, adaptado. Implementado para recuperar valores de conversão de FEM em tensão por meio da interface entre a <i>Raspberry Pi Zero W</i> e o ADS1118. Fonte: Yousef, adaptado.	81
Figura 52 – Código em C, adaptado. Implementado para recuperar valores de conversão de FEM em tensão por meio da interface entre a <i>Raspberry Pi Zero W</i> e o ADS1118. Fonte: Yousef, adaptado.	82
Figura 53 – Código em C, adaptado. Implementado para recuperar valores de conversão de FEM em tensão por meio da interface entre a <i>Raspberry Pi Zero W</i> e o ADS1118. Fonte: Yousef, adaptado.	83

Figura 54 – Código em C, adaptado. Implementado para recuperar valores de conversão de FEM em tensão por meio da interface entre a <i>Raspberry Pi Zero W</i> e o ADS1118. Fonte: Yousef, adaptado.	84
Figura 55 – <i>Script Matlab</i> - Uso da função <i>isoutlier</i> . Fonte: (MATHWORKS, 2020).	85

Lista de tabelas

Tabela 1	– Normas e referências importantes. Fonte: (VOLBRECHT, 2007).	25
Tabela 2	– Principais termopares codificados em letras e suas especificações. Fonte: (BURNS; SCROGER, 1989) e (BURNS et al., 1993).	25
Tabela 3	– Comparando plataformas. Fonte:(FILIPEFLOP, 2020).	30
Tabela 4	– Lista de materiais. * Valor incluso no kit acima. ** Item não precificado. Fonte: O autor.	34
Tabela 5	– Pinos de conexão, interface ADS1118 - <i>Raspberry Pi Zero W</i> . Fonte: O autor.	40
Tabela 6	– Links disponíveis	80

Lista de abreviaturas e siglas

ABNT	Associação Brasileira de Normas Técnicas
ANOVA	<i>Analysis of Variance</i>
ANSI	<i>American National Standards Institute</i>
ARM	<i>Advanced RISC Machine</i>
ADC	<i>Analog to Digital Converter</i>
CJC	<i>Cold-Junction-Compensation</i>
CPU	<i>Central Process Unit</i>
DIN	<i>Deutsches Institut für Normung</i>
EMI	<i>Electromagnetic Emission</i>
FEM	<i>Electromotive force</i>
GPIO	<i>General Purpose Input/Output</i>
GPU	<i>Graphics Processing Unit</i>
HTML	<i>HyperText Markup Language</i>
IEC	<i>International Electrotechnical Commission</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IoT	<i>Internet of Things</i>
IP	<i>Internet Protocol</i>
LCD	<i>Liquid Crystal Display</i>
MISO	<i>Master Input Slave Output</i>
MOSI	<i>Master Output Slave Input</i>
RAM	<i>Random Access Memory</i>
RPi	<i>Raspberry Pi</i>
RTD	<i>Resistance Temperature Devices</i>

SD	<i>Secure Digital</i>
SSH	<i>Secure Shell</i>
SoC	<i>System-on-a-chip</i>
SPI	<i>Serial Peripheral Interface</i>
TI	<i>Texas Instruments</i>
VNC	<i>Virtual Network Computing</i>
Wi-Fi	<i>Wireless Fidelity</i>

Lista de símbolos

bps	bits por segundo
B	bytes
f	frequência [Hz]
G	giga
I	corrente elétrica [A]
k	quilo
μ	micro
m	mili
M	mega
%	por cento
T	temperatura [°C]
t	tempo [s]
V	unidade de tensão elétrica [V]

Sumário

1	INTRODUÇÃO	18
1.1	CONTEXTUALIZAÇÃO	18
1.2	JUSTIFICATIVA	19
1.3	OBJETIVOS	20
1.3.1	Objetivos específicos	20
1.3.2	Requisitos	20
1.3.3	Aspectos Metodológicos	21
1.4	ORGANIZAÇÃO DO TRABALHO	23
2	FUNDAMENTAÇÃO TEÓRICA	24
2.1	REVISÃO BIBLIOGRÁFICA	24
2.2	EMBASAMENTO	28
2.2.1	Aplicação da Eletrônica Digital em sensores termopares	28
2.2.2	A escolha da plataforma empregada no registro e compartilhamento de medidas	29
2.2.3	Estatística aplicada	30
3	DESENVOLVIMENTO	33
3.1	MATERIAIS UTILIZADOS	34
3.1.1	A placa Raspberry Pi Zero Wh	34
3.1.2	Pinos de GPIO da RPi	35
3.1.3	Placa <i>TI 430BOOST-ADS1118</i>	36
3.1.4	Características do módulo <i>TI 430BOOST-ADS1118</i>	36
3.1.5	Sonda do termopar tipo K:	37
3.1.6	Indicador LCD:	38
3.1.7	Placa de adaptação	39
3.2	CONEXÃO DE HARDWARE E INTERFACE	40
3.2.1	Protocolo de comunicação	40
3.2.2	Conexão de hardware	43
3.3	DESCRIÇÃO DE SOFTWARE	44
3.3.1	Configuração - <i>Raspberry Pi</i> e <i>Linux</i>	44
3.3.2	Medição de temperatura	45
3.3.3	Detalhes do processo	46
3.3.4	Levantamento de pontos e possibilidade de realizar calibração	49
3.3.5	Servidor Web	51

4	RESULTADOS	54
4.1	Aferindo a temperatura	54
4.1.1	Desenvolvimento da página <i>web</i>	55
4.1.2	Ensaio de medidas de temperatura utilizando uma resistência aquecida	57
5	ANÁLISE DOS RESULTADOS	60
5.1	Interface <i>web</i>	61
5.2	Análise de dados	61
5.2.1	Eliminação de <i>outliers</i>	61
5.2.2	Análise de variância de medidas repetidas de duas vias (ANOVA)	63
5.2.2.1	ANOVA e o Teste de Normalidade	64
5.2.2.2	ANOVA e o Teste de Esfericidade	65
5.2.3	ANOVA e o teste complementar - <i>Post-Hoc</i>	67
5.3	<i>Bland-Altman</i>	69
6	CONSIDERAÇÕES FINAIS	73
6.1	Proposta de melhorias	74
	REFERÊNCIAS	75
	ANEXOS	78
	ANEXO A – PRIMEIRO ANEXO	79
	ANEXO B – SEGUNDO ANEXO	80
	ANEXO C – TERCEIRO ANEXO	85

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

Recursos de medição e monitoramento de temperatura estão presentes nos mais diversificados setores da indústria, tendo como principal finalidade expressar, de forma quantitativa, o grau de agitação térmica das moléculas que compõem determinados materiais ou substâncias. Segundo Thomazini e Albuquerque ([THOMAZINI; ALBUQUERQUE, 2011](#)), eles geralmente são empregados em sistemas de controle de processos que garantem a qualidade final de produtos, tais como bebidas e alimentos, medicamentos, combustíveis, metais, componentes elétricos e eletrônicos, entre outros.

Para realizar medidas satisfatórias, de acordo com Morris ([MORRIS, 2001](#)), cada situação requer o uso de instrumentos apropriados, dessa forma, a escolha de um dispositivo de medição de temperatura dependerá da aplicação e de suas condições de funcionamento, levando-se em consideração suas características, funcionalidades e custo. O mercado dispõe de diversos modelos de medidores de temperatura, cada qual com seu propósito e limites de operação. Os mais comuns são aqueles utilizados para aferir a temperatura corporal e a de locais como o interior de freezers, fornos e reservatórios.

Nesse contexto, verifica-se a disponibilidade de medidores que não necessitam de contato direto com as substâncias cujas temperaturas se deseja medir, como os pirômetros à radiação total ou parcial e também aqueles que precisam entrar em contato com as substâncias para aferir suas temperaturas, como os termômetros à dilatação de líquido, gás ou sólido, à resistência elétrica e à par termo elétrico - mais conhecidos como termopares ([WILSON, 2005](#)).

Embora os pirômetros utilizem sensores infravermelhos para medir temperaturas elevadas, eles geralmente não são precisos e sensíveis à emissividade das superfícies de interesse. Já os termômetros à dilatação, são mecanismos robustos que medem uma ampla faixa de temperatura, no entanto, por apresentarem um elevado tempo de resposta, não são recomendáveis para leituras transientes. E, enquanto os termoresistores (RTDs) são dispositivos estáveis que cobrem largas faixas de temperatura, os termistores, apesar de mais precisos, medem faixas de temperatura menores ([INSTRUMENTS, 2019b](#)). Ambos não são tão robustos e baratos quanto os termopares e estão sujeitos a imprecisões causadas por efeitos de auto aquecimento ([FIALHO, 2002](#)).

Makris et al. ([MAKRIS et al., 2003](#)) destaca em sua pesquisa que devido a simplicidade, robustez, e capacidade de medir uma ampla faixa de temperatura, dentre outras aplicações, os termopares são dispositivos utilizados nos mais variados experimentos cien-

tíficos, principalmente aqueles que buscam analisar a relação entre as propriedades físicas envolvidas.

Os métodos tradicionalmente empregados para coletar medidas de temperatura são excessivamente trabalhosos, lentos e estão sujeitos a inevitáveis fontes humanas de erros e imprecisões. Como exemplo, pode-se destacar os casos na qual o pesquisador precisa estar próximo ao leitor do instrumento e anotar manualmente os valores lidos. Nesse sentido, o presente estudo visa implementar um dispositivo que conta com processo de armazenamento automático das medidas e permite exibição compartilhada.

A ferramenta desenvolvida é composta por uma interface entre um termopar tipo K e uma *Raspberry Pi*, por meio do módulo *430BOOST-ADS1118*. Esse dispositivo comunica-se com smartphones e computadores pessoais via *Wi-Fi* e possui caráter complementar para aplicações laboratoriais as quais necessitam de medidas específicas de temperatura, facilitando o processo de registro e acompanhamento de dados.

1.2 JUSTIFICATIVA

Quando se trata da obtenção e acompanhamento de dados em experimentos laboratoriais, alguns medidores de temperatura modernos mostram-se muito funcionais. Visto que eles poupam o tempo do pesquisador, permitindo registrar automaticamente e acompanhar, à distância, as medidas realizadas.

Os dispositivos comerciais disponíveis que desempenham estas funções geralmente são equipamentos com maior complexidade e alto custo, em torno de R\$ 4945,00 (OMEGA, 2020). Também existem os tradicionais que, apesar de serem mais acessíveis, não dispõem de recursos de compartilhamento e salvamento automático de dados, como o medidor TM-902C, encontrado por R\$ 128,00.

A motivação deste projeto consiste em complementar os dispositivos básicos de medição já existentes, para que havendo necessidade, implemente-se um medidor de temperatura de baixo custo, empregando uma plataforma *Raspberry Pi* e um módulo *TI 430BOOST-ADS1118* com termopar do tipo K. O dispositivo a ser implementado será de fácil manuseio, apresentará resultados tão precisos quanto os dispositivos comerciais e além disso, o operador poderá acompanhar as medidas de temperatura, via *Wi-Fi*, por meio do seu *smartphone* ou computador pessoal e registrar, automaticamente, tais valores e seus respectivos horários.

Todo o processamento será realizado pela *Raspberry Pi* e por meio de um cartão microSD, os valores medidos poderão ser armazenados em um banco de dados, mesmo se ocorrerem problemas de conexão de rede que impeçam o armazenamento em nuvem ou o *upload* e *download* de dados. Desta forma, será viabilizada uma maior comodidade

e praticidade ao operador, garantindo a economia do tempo de permanência no laboratório onde se encontra o instrumento de medição e do tempo que seria gasto anotando manualmente os valores aferidos.

1.3 OBJETIVOS

O principal objetivo deste trabalho é complementar as ferramentas de monitoramento e registro de temperatura já existentes, implementando um dispositivo de baixo custo, baseado em uma interface entre a placa *Raspberry Pi*, o módulo *430BOOST-ADS1118* e um termopar tipo K, com a finalidade de empregá-lo em experimentos laboratoriais que envolvam medidas e análise da variação de temperatura.

1.3.1 Objetivos específicos

A construção deste trabalho foi realizada com base nos seguintes objetivos:

- Desenvolvimento de interface de *hardware* e *software* entre a *Raspberry Pi* e o módulo *430BOOST-ADS1118* para exibição dos valores medidos no *display* de LCD;
- Realização de medidas confiáveis de temperatura, utilizando um termopar tipo K, na faixa de operação estabelecida: 25 a 505 °C;
- Acesso facilitado, através da exibição e registro dos valores medidos, além da definição de intervalos de tempo entre medidos, via *Wi-Fi*, por meio de um *smartphone* ou computador pessoal;
- Armazenamento, dos valores de temperaturas medidas e horários, em planilhas Excel, para gerar um banco de dados disponível no cartão microSD.

1.3.2 Requisitos

Os principais requisitos para a realização deste estudo foram:

- Os sinais capturados por meio do termopar tipo K devem ser tratados utilizando o conversor Analógico-Digital do módulo *430BOOST-ADS1118*;
- O dispositivo deve ser implementado utilizando a tabela de medidas da ITS-90 (NIST, 2014) e suas medidas devem ser comparadas com as medidas obtidas por meio de um instrumento comercial de referência (TM-902C);
- Um *data logger* deve ser utilizado para realizar o salvamento automático dos dados de interesse em uma planilha;

- A conexão *Wi-Fi* disponível na *Raspberry Pi* deve ser empregada para transmitir as informações por meio de uma arquitetura cliente-servidor criada para facilitar o acesso e salvamento dos dados, utilizando um navegador;
- A página *web* deve dispor de campos com opções para digitar o intervalo de tempo entre as medidas a qual se deseja obter, iniciar e finalizar o processo de medidas com salvamento automático na memória externa e outra alternativa, para realizar o *download* de um arquivo no formato ".csv", a qual conterá os dados registrados em uma planilha.

1.3.3 Aspectos Metodológicos

Tomando como base procedimentos apresentados por Gil (GIL, 2002), a metodologia deste trabalho compreendeu as seguintes etapas:

- Plano de trabalho: Foram definidos o escopo, a justificativa, o referencial teórico, os objetivos e os requisitos do projeto;
- Implementação de *hardware*: Foi separada em módulos individuais para facilitar os testes e análises. Durante a implementação, a análise do funcionamento de cada módulo foi realizada, consultando-se documentos disponibilizados pelos seus respectivos fabricantes. Estes módulos consistem basicamente em um sensor termopar tipo K, uma placa *430BOOST-ADS1118* com multiplexador, conversor analógico-digital (AD), bloco de transmissão de dados SPI, *display* de LCD e a placa *Raspberry Pi zero W*, que conta com diversos recursos, dentre eles um processador ARM, pinos GPIOs e comunicação via *Wi-Fi*. Em seguida, foi realizada a integração destes módulos por meio da comunicação SPI, utilizando-se uma placa adaptadora;
- Validação de *hardware*: Foram realizados diversos testes por meio da comunicação entre computador pessoal e *Raspberry Pi* utilizando-se comandos e verificando os resultados via terminal do Sistema Operacional *Raspbian*. Nesta etapa, sinais analógicos de tensão foram capturados, analisados e tratados resultando em valores digitais medidos por meio do termopar tipo K. Posteriormente, foi realizado um levantamento de pontos em intervalos regulares de temperatura comparando-se os valores obtidos por meio do *430BOOST-ADS1118* com as medidas provenientes do dispositivo de referência TM-902C e foram ajustados, via *software*, os valores estabelecidos na tabela de medidas da ITS-90;
- Desenvolvimento de *Software* e interface gráfica: Foram realizadas as configurações da *Raspaberry Pi*, bem como a adaptação de um código em C, disponibilizado pela *Texas Instruments*, para fazer a interface entre o ADS1118 e a RPi, converter e

valores ADC para valores de temperatura reais e adicionar a capacidade de transferência de dados para registro (*data logger*) e exibição. Outra parte desenvolvida foi uma página destinada ao acompanhamento das medidas em navegadores web, via *Wi-Fi*. Esta etapa envolveu o uso de *framework web* baseado em processos *back-end* e *front-end*, na qual, para a implementação da interface gráfica da página web, utilizou-se, como referência, modelos escritos em *JavaScript* e HTML;

- Validação da interface gráfica: A página web foi testada várias vezes, utilizando-se os campos destinados ao operador do dispositivo: Escolha de modo (gravação ou acompanhamento de medidas), período programado (digitar o intervalo de tempo entre as medidas a qual se deseja obter), iniciar ou finalizar o registro de dados e a função que realiza o *download* do arquivo em formato .csv;
- Validação do processo de medidas: Foram realizadas medidas utilizando-se uma resistência de fogareiro elétrico WY-01B. Este permitiu regular a temperatura em intervalos de 15 °C, na faixa de 25 a 505 °C. Em cada valor fixado, a temperatura foi medida três vezes, simultaneamente, por dois instrumentos independentes: um dispositivo comercial (TM-902C) e o dispositivo implementado neste projeto, ambos utilizando termopar tipo K.

Foi realizada uma análise quantitativa com o objetivo comparar os valores resultantes das medidas pontuais de temperatura na faixa estabelecida. Estas medidas, obtidas por cada termopar, foram classificadas em grupos:

Grupo 1: Medidas de referência, oriundas do TM-902C.

Grupo 2: Medidas provenientes do módulo *430BOOST-ADS1118*.

As medidas do grupo 2 foram registradas automaticamente, enquanto as medidas do grupo 1 foram coletadas de forma manual. Ambas foram salvas em planilhas.

Como existe uma correlação entre as medidas pontuais, e por se tratar de variáveis numéricas estatisticamente dependentes, medidas por meio de dispositivos distintos, optou-se por realizar a Análise de Variância de Medidas Repetidas (ANOVA) de duas vias, a qual permitiu verificar se existe ou não diferença entre os grupos de medidas. Fazendo-se necessário, em seguida, a complementação desta análise, que se deu por meio do teste de *Sidak*, cuja análise permite identificar entre quais grupos tal diferença ocorreu.

Por fim, foi avaliada a concordância, entre as metodologias utilizadas, pelos dispositivos comparados, que medem o mesmo parâmetro, a temperatura. Isto foi possível por meio do teste de *Blend-Altman*, utilizando um intervalo de confiança de 95%.

Neste estudo, o ambiente de análise foi o *software* SPSS®;

- Documentação: Elaboração do documento contendo todo o desenvolvimento, resultados e considerações finais do projeto e apresentação do dispositivo implementado.

1.4 ORGANIZAÇÃO DO TRABALHO

O **capítulo 2** apresentará uma breve revisão revelando contribuições referentes aos princípios de funcionamento de um termopar, assim como serão abordados aspectos relacionados à aplicação da eletrônica digital em dispositivos termopares e a escolha da plataforma empregada no registro e compartilhamento de medidas via comunicação *Wi-Fi*.

Em seguida, os **capítulos 3 e 4** tratarão da metodologia empregada no desenvolvimento deste projeto, abrangendo as características dos materiais utilizados, implementação de *hardware* e *software*, bem como os resultados obtidos.

Por fim, a análise dos resultados deste estudo serão descritos no **capítulo 5**, seguido das considerações finais no **capítulo 6**, revelando-se a possibilidade de trabalhos futuros que podem tornar o dispositivo mais eficiente. Ainda, os anexos mostrarão material complementar relevante.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 REVISÃO BIBLIOGRÁFICA

A conversão de calor em eletricidade foi descoberta em 1821 pelo físico estoniano Thomas Johann Seebeck. Por meio de seus estudos experimentais, ele observou que em um circuito fechado, formado por dois condutores metálicos A e B, uma Força Eletromotriz (FEM) era gerada ao variar a temperatura nas regiões de junção (CHANDLER, 2010). A Figura (1) apresenta o circuito utilizado por Seebeck, na qual quando a temperatura de junção T1 fica maior que T2, uma corrente elétrica começa a fluir.

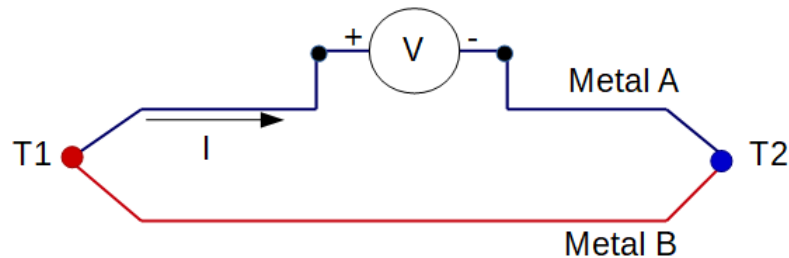


Figura 1 – Esquema representativo do circuito implementado por Seebeck. Fonte: O autor.

De acordo com Thomazini e Albuquerque (THOMAZINI; ALBUQUERQUE, 2011), a FEM gerada está relacionada com o campo elétrico formado por meio do aquecimento - que é função do gradiente de temperatura na região de interesse. Segundo a Teoria de Seebeck, a FEM é dada pela “Eq. (2.1)” a seguir:

$$V = \int_{T_1}^{T_2} (S_B(T) - S_A(T)) dT \quad (2.1)$$

Onde S_A e S_B são os coeficientes de Seebeck dos metais conectados e T1 e T2 são as temperaturas das respectivas junções. Os coeficientes de Seebeck não são lineares e dependem da temperatura absoluta e da estrutura molecular dos condutores (POLLOCK, 1991). Quando estes coeficientes são constantes em uma dada faixa de temperatura, a “Eq. (2.1)” passa a ser reescrita conforme a “Eq. (2.2)” abaixo:

$$V = (S_B - S_A) \cdot (T_2 - T_1) \quad (2.2)$$

Com base em estudos experimentais voltados ao efeito termoelétrico, o Efeito Seebeck é uma manifestação física que, dentre outras utilizações, é muito empregada em

medidas de largas faixas de temperatura por meio de sensores termopares.

BURNS et al. (BURNS et al., 1993) afirmam que um termopar pode ser construído por diferentes combinações de metais, entretanto, em aplicações reais são utilizadas combinações normalizadas, tendo em vista que cada condutor possui tolerâncias e capacidades próprias as quais limitam a precisão das medidas. Em processos de calibração, pode-se consultar a Escala Internacional de Temperaturas de 1990 (ITS-90), onde são apresentados valores de temperaturas, nas unidades kelvin e grau Celsius, bem como as respectivas tensões de saída previsíveis para cada tipo de termopar.

Foram criadas normas técnicas internacionais para manter e padronizar diversos itens, como os sensores termopares. Para diferenciar os tipos de termopares, as cores de revestimento são designadas e codificadas de acordo com o padrão estabelecido nas normas utilizadas em cada país. Muitas especificações de termopares podem ser encontradas nas Normas Técnicas Brasileiras estabelecidas pela ABNT (Associação Brasileira de Normas Técnicas) bem como em Normas Internacionais (ISO). A Tabela (1) apresenta algumas Normas Internacionais relevantes que podem ser utilizadas como referência:

Organização	Origem	Norma
ABNT	Brasil	NBR 12771
ANSI	Estados Unidos	ANSI MC 96.1
BS	Inglaterra	BS 4937
DIN	Alemanha	DIN 43710
JIS	Japão	JIS C1602
IEC	Internacional	IEC 60584-1:2013

Tabela 1 – Normas e referências importantes. Fonte: (VOLBRECHT, 2007).

A Tabela (2) indica alguns dos sensores termopares mais encontrados no mercado, com especificações de acordo com a norma americana ANSI MC 96.1.

Tipo	Material	Faixa de Op. [°C]	Sensibilidade [$\mu V/^{\circ}C$]
B	Pt30%Rh-Pt6%Rh	0 a 1820	7,7
E	Cr-Co	-270 a 1000	76
J	Fe-Co	-210 a 1200	55
K	Cr-Al	-270 a 1370	39
N	Nicrosil-Nisil	-200 a 1200	68
R	Pt13%Rh-Pt	-50 a 1760	11,7
S	Pt10%Rh-Pt	-50 a 1760	10,4
T	Cu-Co	-270 a 400	45

Tabela 2 – Principais termopares codificados em letras e suas especificações. Fonte: (BURNS; SCROGER, 1989) e (BURNS et al., 1993).

Em aplicações práticas, a compatibilidade de equipamentos pode determinar, por exemplo, o diâmetro da sonda de um termopar. Levando-se em consideração os estudos realizados por Burns e Scroger (BURNS; SCROGER, 1989), torna-se evidente que é possível evitar diversos problemas e falhas em serviços de medição de temperatura, estabelecendo-se critérios durante a seleção de um termopar, tais como: faixa de temperatura, resistências química, à vibração e à abrasão e requisitos de instalação do termopar. E também, realizando a sua calibração por meio do método da comparação. Desta forma, os valores obtidos utilizando outros instrumentos de precisão podem ser adotados como referência, permitindo ajustar os parâmetros correspondentes para o funcionamento adequado do termopar.

Dependendo dos materiais que compõem os fios empregados em um termopar, a faixa de temperatura medida pode variar de -200 a 1820 $^{\circ}\text{C}$. A Figura (2) apresenta um gráfico, com a relação entre FEM [mV] e T $^{\circ}\text{C}$, dos termopares tipo J, K e S.

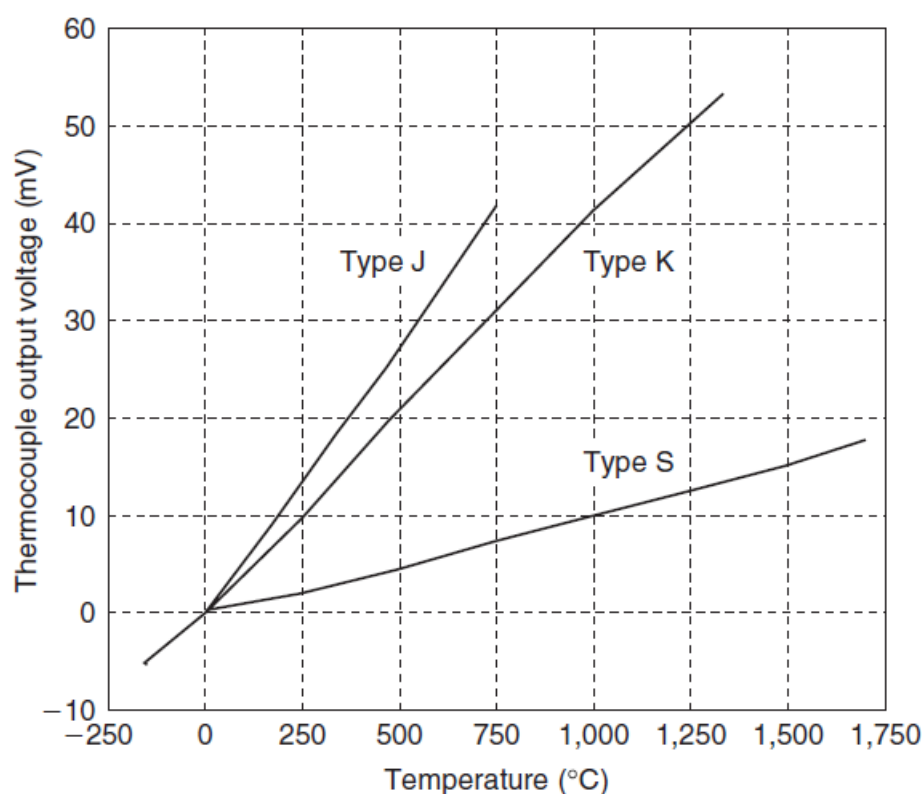


Figura 2 – Faixa de trabalho dos termopares tipo J, K e S. Tensão x Temperatura. Fonte: (ZUMBALEN, 2008).

A Figura (3) apresenta o esquema de um termopar do tipo K, nele os dois condutores estão ligados em uma de suas extremidades - ponto de junção T_h - e foram mantidos abertos na outra extremidade, conhecida como junção de compensação fria (CJC) - ponto de junção T_c . Quando há diferença entre as temperaturas dessas duas junções, surge na junção de referência uma FEM de origem térmica, que é proporcional à diferença de

temperatura entre as junções.

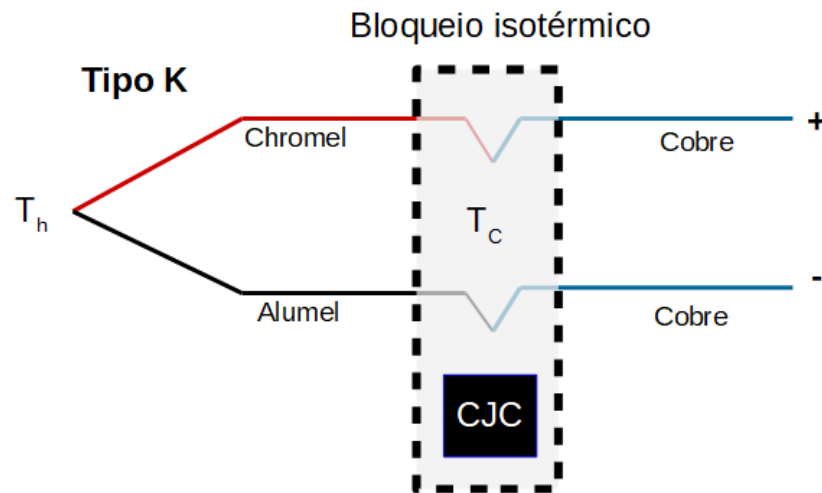


Figura 3 – Esquema de um Termopar tipo K. Fonte: Texas Instruments, adaptado.

Essa FEM é medida em milivolts [mV] e pode ser convertida para temperatura, em graus Celsius [°C], por meio do Coeficiente de Seebeck. A Figura (4) apresenta um gráfico com a relação entre Coeficiente de Seebeck [$\mu\text{V}/^\circ\text{C}$] e Temperatura [°C] dos termopares tipo J, K e S.

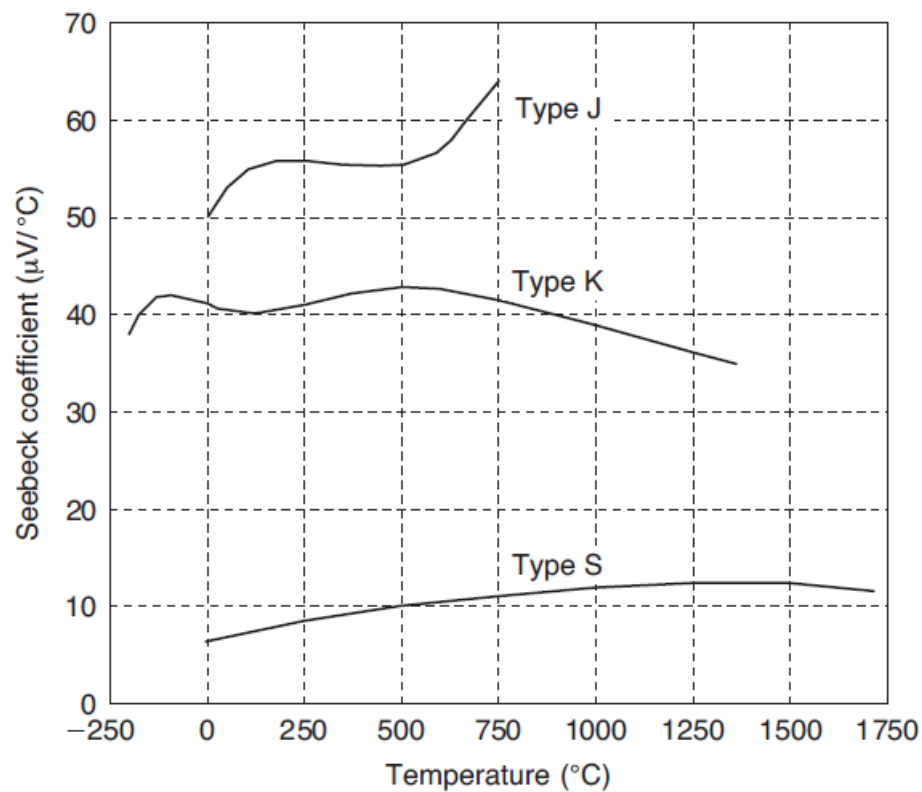


Figura 4 – Sensibilidade dos termopares tipo J, K e S. Coeficiente de Seebeck x Temperatura. Fonte: (ZUMBAHLEN, 2008).

2.2 EMBASAMENTO

2.2.1 Aplicação da Eletrônica Digital em sensores termopares

Em conjunto com o desenvolvimento e compartilhamento de *softwares*, a eletrônica digital viabilizou novas aplicações, abrindo caminho à inúmeras possibilidades, tanto no âmbito acadêmico quanto no industrial, principalmente com a implementação de sistemas de medição que utilizam Internet das Coisas (IOT) e o aperfeiçoamento das partes que os constituem. Segundo Webster ([WEBSTER, 2000](#)) e Aguirre ([AGUIRRE, 2019](#)), a representação de medidas na forma digital ocorre por meio do processo de discretização e da conversão analógico/digital (A/D) dos sinais referentes à variáveis cuja natureza é contínua em função do tempo, utilizando-se sensores e transdutores.

Em aplicações práticas, para medir as grandezas desejadas, instrumentos como os multímetros digitais podem dispor de diversas funções selecionáveis, tais como voltímetro, amperímetro, ohmímetro, capacitímetro e indutímetro. Além destas funções, muitos dispositivos permitem realizar medidas de temperatura por meio de sondas plugáveis de termopares. Como exemplo, pode-se citar a placa *430BOOST-ADS1118* da *Texas Instruments* (TI), a qual embora tenha sido desenvolvida para funcionar acoplada à plataforma *LaunchPad MSP430*, dependendo da necessidade, pode ser adaptada em outras.

Em processos de monitoramento de temperatura, muitos dispositivos modernos realizam o tratamento de sinais por meio de placas integradas, contando com microprocessadores, microcontroladores e diversos periféricos embutidos, como: memória, conversores A/D, barramentos de comunicação, entre outros, aos quais são programados para realizar a calibração e a exibição das medidas em *displays* de LCD. A Figura (5) apresenta três exemplos de plataformas que podem ser empregadas no desenvolvimento de dispositivos de medição de temperatura.

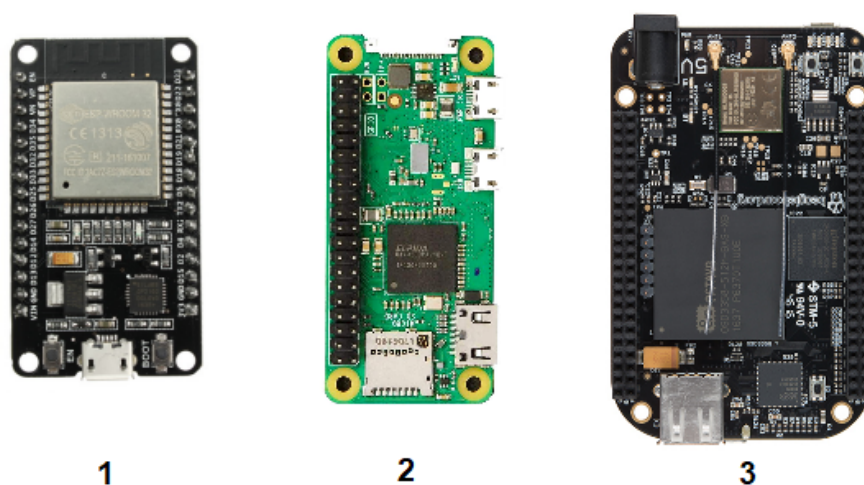


Figura 5 – Plataformas: 1. *ESP32*, 2. *Raspberry Pi Zero Wh* e 3. *BeagleBone Black Wireless*. Vistas superiores. Fonte: ([FILIPEFLOP, 2020](#)), adaptado.

2.2.2 A escolha da plataforma empregada no registro e compartilhamento de medidas

Conforme apresentado por Oliveira et al. (OLIVEIRA et al., 2018), o acesso e monitoramento de medidas de temperatura pode ser realizado por meio de tecnologias *wireless*, as quais embora estejam suscetíveis a fontes de interferência e a limitações de alcance, permitem alta velocidade no tráfego de dados e um maior grau de mobilidade ao pesquisador durante o acompanhamento das medidas, facilitando operações e economizando tempo e recursos.

Tendo em vista a relação direta entre o escopo do projeto e a seleção da plataforma a ser utilizada, nesta etapa foram levados em consideração os requisitos estabelecidos para o desenvolvimento de uma arquitetura cliente-servidor dedicada ao acompanhamento remoto das medidas de temperatura por meio de uma página web. E a implementação de um *data logger*, para gerar um banco de dados e salvar as medidas independentemente da estabilidade da rede de transmissão sem fio, evitando-se a perda de dados em casos de eventuais problemas com a conexão de rede. Vale enfatizar que foi empregado o protocolo de rede *Wi-Fi*, a qual destaca-se por transmitir dados entre longas distâncias e em alta velocidade.

Segundo Rivera (RIVERA, 2010), o padrão de rede *Wi-Fi*, estabelecido pela arquitetura IEEE 802.11n, permite a comunicação entre dispositivos móveis via ondas de rádio e é amplamente utilizado em aplicações tradicionais que operam na faixa de 2,4 GHz. Isto ocorre por meio de roteadores, aos quais analisam os destinos dos sinais tratados e decidem suas rotas de transmissão. Esta conexão de baixa potência apresenta um alcance variável, de 30 a 100 metros, considerado adequado para redes locais, com uma taxa de transmissão de dados entre 65 e 600 Mbps. Tal variação depende de fatores que influenciam a operação, como: tipo e potência da antena, ambiente (interno ou externo), fontes de interferência, entre outros.

Em redes *Wi-Fi*, pontos de acesso e repetidores podem ser empregados para aumentar a área de alcance do sinal transmitido pelo roteador. Moraes (MORAES, 2010) recomenda o uso da rede *Wi-Fi* para comunicações sem fio, devido às vantagens proporcionadas quando comparada com o protocolo *Bluetooth*, em termos de área coberta pelo sinal e taxa de transferência de dados.

Em face do que foi exposto, as plataformas apresentadas na Fig. (5) foram comparadas. Em análise, verificou-se que as três permitem conexão à rede sem fio, entretanto, apesar da *ESP32* ter conexão *Wi-Fi*, frequência de operação de 160 MHz e ser de baixo custo, ela não conta com uma memória *flash* (não volátil) integrada suficiente para implementar um *data logger*. E apesar da *BeagleBone Black Wireless* dispor de comunicação *Wi-Fi*, frequência de operação e memória *flash* satisfatórias, seu custo é muito elevado.

Portanto, optou-se por utilizar a *Raspberry Pi Zero*, uma plataforma que, de acordo com Juca e Pereira ([JUCA; PEREIRA, 2018](#)), tem custo razoável, permite expandir a memória *flash*, por meio do seu *slot* para cartão microSD (*Secure Digital*) e processar tarefas complexas, como monitoramento online. A Tabela (3) apresenta algumas características das três plataformas comparadas:

Plataforma:	ESP	Raspberry Pi	BeagleBone Black
Versão:	ESP32	RPi Zero Wh	BBB Wireless
Sistema Op.:	Mongoose OS	Linux, Raspbian	Linux, Android
Linguagem de Prog:	C, Python	C, Python	C, Python
Contr./Processador:	Tensilica Xtensa LX6	BCM2835 ARM	OSD3358 ARM
Frequência de Op:	240 MHz	1 GHz	1 GHz
Memória RAM:	520 kB	512 MB	512 MB
Armazenamento:	4MB	Slot MicroSD	4GB, Slot MicroSD
Conexão de rede:	Wi-Fi 802.11n	Wi-Fi 802.11n	Wi-Fi 802.11n
Pinos GPIOs:	36	40	92
Pinos ADC:	18	-	7
Preço em R\$:	89,90	139,90	799,90

Tabela 3 – Comparando plataformas. Fonte: ([FILIPEFLOP, 2020](#)).

É importante enfatizar que embora a plataforma escolhida não tenha conversor analógico digital, este está incluso no módulo *430BOOST-ADS1118*.

2.2.3 Estatística aplicada

Na pesquisa experimental geralmente se utiliza a análise estatística. O desenvolvimento das técnicas estatísticas tem sido notável e sua aplicabilidade na pesquisa experimental tão adequada que não se pode hoje deixar de utilizá-las no processo de análise dos dados ([GIL, 2002](#)).

Em análises estatísticas podem ser utilizados diferentes métodos, como a Análise de Variância (ANOVA), que consiste em testes baseados em hipóteses que permitem realizar comparações entre três ou mais grupos de dados. Existem diferentes testes ANOVA, como o teste ANOVA de uma via, que permite comparar a variância nas médias de um conjunto de dados amostrados, em casos que se tenha apenas uma variável independente. E também a ANOVA de duas vias que possibilita a comparação das variâncias entre as médias de três ou mais conjuntos de dados separadas em dois grupos independentes, com a finalidade de examinar o efeito de dois fatores em uma variável dependente, como a temperatura.

Em uma análise de dados por meio do método ANOVA de duas vias, é necessário que: a variável dependente seja contínua (medida em uma escala que pode ser subdividida em incrementos, como a temperatura, em °C), cada amostra seja obtida de forma

independente das demais e que as duas variáveis independentes estejam separadas em grupos categóricos, como: “medidas do Grupo 1 – TM-902C ” e “medidas do Grupo 2 – ADS1118”. Além disso, por meio do *software* IBM SPSS, algumas premissas podem ser verificadas utilizando-se testes com intervalo de confiança de 95% e nível de significância estatística (p) de 5%, tais como:

- **Teste de Normalidade:** verifica se as amostras extraídas de uma população estão normalmente distribuídas, conforme apresentado por Miot ([MIOT, 2017](#));
- **Teste de Esfericidade:** são calculadas as diferenças entre cada par de amostras dos conjuntos de medidas repetidas e, em seguida, a diferença pontual entre as variâncias. Realiza-se a suposição de que a matriz de variância-covariância da variável dependente tenha forma circular ou esférica ([IBMKNOWLEDGECENTER, 2019](#)));
- **Teste complementar - *Post-Hoc*:** permite analisar diferenças entre os grupos e conjuntos de amostras e verificar se o efeito de interação é significativo ([IBMKNOWLEDGECENTER, 2019](#)).

Nesta metodologia, o valor de significância (p) pode ser usado para determinar a significância estatística em um teste de hipótese, com probabilidade de erro de 5%. Quando este valor é menor que 0,05, dependendo da convenção adotada, a hipótese para os testes pode ser válida, afetando as conclusões obtidas na análise.

Para satisfazer as premissas supracitadas, faz-se necessário descartar eventuais dados discrepantes (*outliers*) dos conjuntos de amostras.

Nesse contexto, a função *isoutlier*, disponível no *software* MATLAB, permite analisar matrizes de dados e verificar a existência de elementos discrepantes, identificando-os de forma prática. Esta função indica a localização de cada *outlier* retornando o valor lógico 1 na matriz de saída. Esta matriz de saída é uma matriz lógica cujos elementos são verdadeiros (1) quando um *outlier* é detectado no elemento correspondente. Por padrão, um valor atípico é um valor que está a mais de três desvios absolutos em escala de distância da mediana ([MATHWORKS, 2020](#)).

Outra metodologia de análise é o teste de *Bland-Altman*, que permite verificar se existe concordância entre dois métodos distintos, aos quais devem medir a mesma quantidade, conforme apresentado por Hirakata e CAMEY ([HIRAKATA; CAMEY, 2009](#)). Logo, este processo de análise avalia se existe concordância entre dois métodos que embora possam partir do mesmo princípio fundamental, utilizam caminhos e meios diferentes para alcançar o mesmo resultado.

Esta análise de concordância entre as médias das medidas provenientes dos dois métodos, pode ser utilizada para estimar os limites do intervalo de confiança, por meio

da “Eq. (2.3)” e, também para relacionar o viés de proporção das medidas e em muitos casos, realizar correções nos dados.

$$L_{S,I} = \pm 1,96.DP \pm \bar{x}_{diff} \quad (2.3)$$

Onde $L_{S,I}$ refere-se aos limites superior e inferior, \bar{x}_{diff} é a média das diferenças e DP é o desvio padrão, apresentado na “Eq. (2.4)”

$$DP = \sqrt{\sum \frac{(x_i - \bar{x})^2}{N}} \quad (2.4)$$

Sendo que x_i corresponde as amostras, \bar{x} é a média da população e N é o tamanho da população.

3 DESENVOLVIMENTO

Neste capítulo, foram apresentados os principais equipamentos, acessórios e procedimentos utilizados na elaboração e desenvolvimento do protótipo do dispositivo de monitoramento e medição de temperatura.

O projeto em questão visa apresentar, de forma simples, um instrumento capaz de medir, salvar e compartilhar valores precisos de temperatura, podendo servir como base para sistemas que envolvam IOT. O seu desenvolvimento foi realizado com base no diagrama apresentado na Fig. (6):

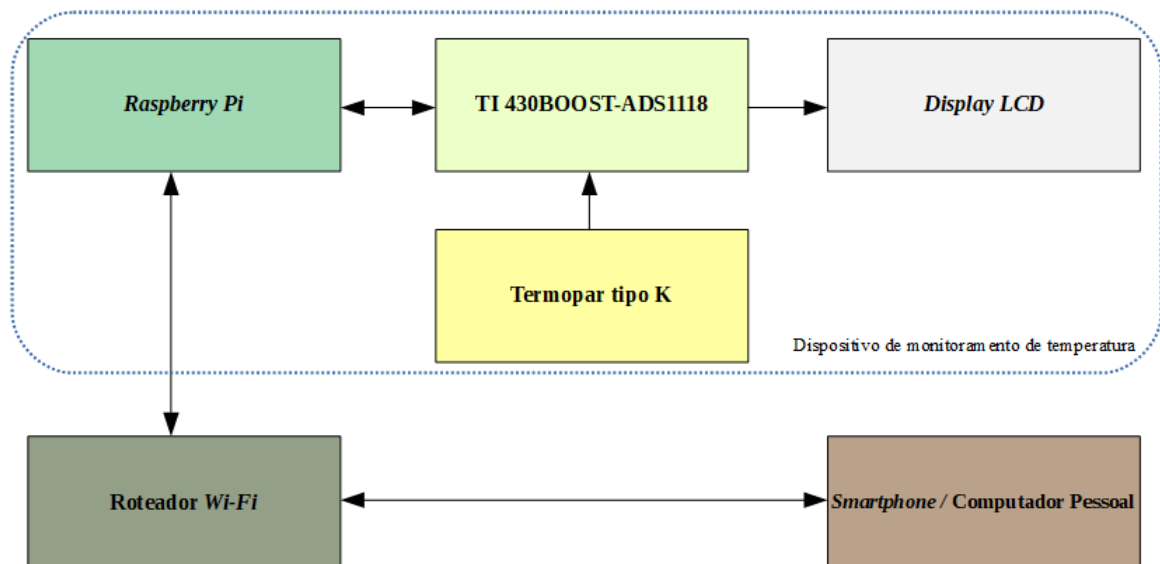


Figura 6 – Diagrama de blocos funcional do projeto - Fluxo de dados. Fonte: O autor.

Este diagrama (Fig. 6) apresenta um sistema de comunicação dividido em duas partes principais, o instrumento medidor (estrutura a qual contém os módulos de aquisição e exibição) e o controle remoto (para envio de comandos e obtenção das informações desejadas). Ambas comunicam-se por intermédio de uma rede *Wi-Fi* utilizando a arquitetura cliente-servidor.

A parte do instrumento de monitoramento de temperatura é composta por um detector (termopar do tipo K), um transdutor (dispositivos intermediários de transferência de dados, formado por um módulo *430BOOST-ADS1118* e uma *Raspberry Pi*) e dispositivo de saída ou indicação (*display LCD*). Enquanto a parte do controle remoto, consiste basicamente em um smartphone ou computador pessoal que permita o acesso a um navegador *web*.

3.1 MATERIAIS UTILIZADOS

Os principais componentes utilizados neste projeto estão listados na Tab.(4):

Item	Modelo/Especificação	Quantidade	Preço R\$
Kit Microcomputador	Placa Raspberry Pi Zero Pi Wh	1	220,00
Cartão de memória	Micro SD 16GB / Classe 10	1	*
Dissipador de calor	Alumínio / 14x14x5 mm	1	*
Fonte de alimentação	5V / 2A DC	1	*
Placa PCB universal	Pré-furada/6x10 cm	1	3,00
Soquete	p/2x20 pinos / 2,54mm	1	1,20
Pinos Conectores	10 pinos / 2,54mm	2	1,50
Termômetro Digital	Simplex / TM-902C	1	**
Kit Texas Instruments	Placa 430BOOST-ADS1118	1	100,00
Display LCD	NHD-C0216CZFSW / SPI	1	*
Sonda Termopar	Tipo K/ 14 AWG (Ø 1,63 mm)	2	*
Computador pessoal	Notebook Samsung AtivBook2	1	**
Roteador Wi-Fi	-	1	**
Fogareiro Elétrico	WY-01B	1	**

Tabela 4 – Lista de materiais. * Valor incluso no kit acima. ** Item não precificado. Fonte: O autor.

O custo total estimado para reprodução de deste projeto gira em torno de R\$ 325,70, levando-se em conta a aquisição de um *kit Raspberry Pi Zero Wh* com acessórios básicos, um módulo *430BOOST-ADS1118* com termopar tipo K e uma placa adaptadora para conectá-los.

3.1.1 A placa Raspberry Pi Zero Wh

A *Raspberry Pi Zero Wh*, apresentada na Fig.(7), é uma placa de baixo custo e com tamanho reduzido (65 x 30 x 5mm). Ela conta com *Wi-Fi* 802.11n e *Bluetooth Low Energy* (BLE) 4.1 integrados. O seu processador, GPU Dual Core VideoCore IV, é baseado no *system on a chip* (SoC) Broadcom BCM2835 ARM11 *single-core* de 1GHz, a qual junto à memória de 512MB, permite que aplicações sejam realizadas em seus 40 pinos de GPIO. Além disso, ela tem *slot* para cartão microSD, conector CSI para câmera, saída de vídeo mini HDMI, 2 portas USB (1 para dados e outra para alimentação) e roda diversas distribuições *Linux*, como o *Raspbian* e o *Ubuntu*. Este microcomputador é alimentado por uma fonte de 5V/2A.



Figura 7 – *Raspberry Pi Zero W*, vistas superior e inferior. Fonte: (FOUNDATION, 2017).

3.1.2 Pinos de GPIO da RPi

Os pinos de entrada/saída de uso geral (GPIO), permitem a comunicação entre componentes eletrônicos conectados. Eles apresentam diferentes características, alguns canais fornecem sinal de tensão constante, de 5 ou 3,3 volts. Outros são pinos de aterramento (GND), usados para fechar um circuito. Também existem aqueles que podem ser programados. Estes permitem alteração do estado de funcionamento, possibilitando ligar ou desligar componentes. Assim, um *firmware* pode ser usado para manipular sinais de entrada e saída dos pinos. A Figura (8) apresenta uma referência da barra de pinos da RPi.

GPI0#	NAME		NAME	GPI0#
	3.3 VDC Power	1	5.0 VDC Power	2
8	GPIO 8 SDA1 (I2C)	3	5.0 VDC Power	4
9	GPIO 9 SCL1 (I2C)	5	Ground	6
7	GPIO 7 GPCLK0	7	GPIO 15 Tx0 (UART)	15
	Ground	9	GPIO 16 Rx0 (UART)	16
0	GPIO 0	11	GPIO 1 PCM_CLK/PWM0	1
2	GPIO 2	13	Ground	14
3	GPIO 3	15	GPIO 4	4
	3.3 VDC Power	17	GPIO 5	5
12	GPIO 12 MOSI (SPI)	19	Ground	20
13	GPIO 13 MISO (SPI)	21	GPIO 6	6
14	GPIO 14 SCLK (SPI)	23	GPIO 10 CE0 (SPI)	10
	Ground	25	GPIO 11 CE1 (SPI)	11
30	SDA0 (I2C ID EEPROM)	27	SCL0 (I2C ID EEPROM)	31
21	GPIO 21 GPCLK1	29	Ground	30
22	GPIO 22 GPCLK2	31	GPIO 26 PWM0	26
23	GPIO 23 PWM1	33	Ground	34
24	GPIO 24 PCM_FS/PWM1	35	GPIO 27	27
25	GPIO 25	37	GPIO 28 PCM_DIN	28
	Ground	39	GPIO 29 PCM_DOUT	29

Figura 8 – Barra de pinos - *Raspberry Pi Zero W GPIO Header*. Fonte: (PI4J, 2012).

3.1.3 Placa *TI 430BOOST-ADS1118*

Devido a sua arquitetura com conversor analógico-digital (AD), precisão e baixo custo, optou-se por utilizar a placa de desenvolvimento *430BOOST-ADS1118* da *Texas Instruments*, apresentada na Fig. (9).



Figura 9 – Módulo medidor de temperatura da *Texas Instruments*, de um canal, juntamente com a sonda termopar tipo K. Fonte: (INSTRUMENTS, 2019a).

3.1.4 Características do módulo *TI 430BOOST-ADS1118*

Este módulo, além de conversor analógico/digital (A/D), inclui um conector para termopares do tipo K e um *Liquid Crystal Display* de 16x2. Sua placa é caracterizada por ser capaz de operar em ambientes na faixa de -40°C a 125°C , dispor de 16 *bits* de resolução, oferecer flexibilidade de modelagem em experimentos, apresentar uma resposta rápida e boa taxa de atualização. O diagrama simplificado deste módulo pode ser visto na Fig. (10).

Nesta placa, a integração do sensor de temperatura, oscilador de 1 MHz, filtro digital e dentre outros componentes, multiplexador e amplificador de ganho de tensão programáveis, faz do ADS1118 um conversor de sinais apropriado para aplicações que abrangem a leitura de temperatura. A descrição de configuração e campos de registro do circuito apresentado na Figura 10 pode ser verificada no Anexo A.

Em contraste com termistores e termoresistores (RTDs), o uso de termopares geralmente simplifica a implementação, utilizando-se circuitos que não requerem excitação. E por basearem-se no Efeito Seebeck, geram tensão elétrica em função da temperatura, sem necessidade de circuito ativo, exigindo-se apenas uma tensão estável de referência e valores de compensação de junção fria.

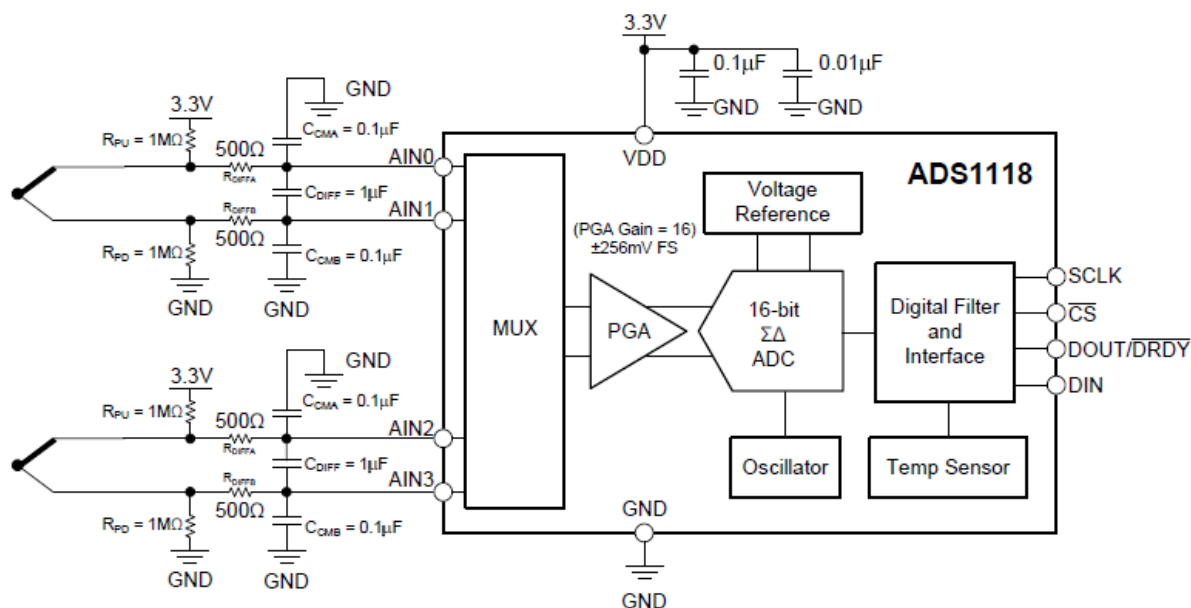


Figura 10 – Diagrama do circuito ADS1118 com canal duplo. Este circuito permite realizar registros por meio da configuração do multiplexador e de outros componentes embutidos no ADS1118. O multiplexador permite a leitura na entrada analógica e o envio de sinais, que podem ser amplificados, ao conversor analógico-digital. Estes sinais são filtrados e transmitidos à RPi. Fonte: (BECKMAN; CHIOYE, 2011).

3.1.5 Sonda do termopar tipo K:

O termopar tipo K consiste em uma extensão de dois fios, formados por condutores diferentes, soldados em uma extremidade, como mostra a Fig. (11). Ele é um dos mais utilizados, por ser capaz de operar em uma larga faixa de temperatura, possuir custo relativamente baixo e possibilitar uma ampla gama de aplicações. É caracterizado por cobrir faixas de temperatura que variam entre -200 °C e 1370 °C, produzir FEM no intervalo de -6,458 mV a 48,838 mV e detectar alteração do sinal elétrico de saída com sensibilidade que chega a aproximadamente 41 $\mu V/^{\circ}C$.

Sua junção é composta por dois termoelementos:

- **O termoelemento positivo (KP)** formado por Cromel (ligas de Ni90% e Cr10%) e,
- **O termoelemento negativo (KN)** formado por Alumel (ligas de Ni95%, Mn2%, Si1% e Al2%).

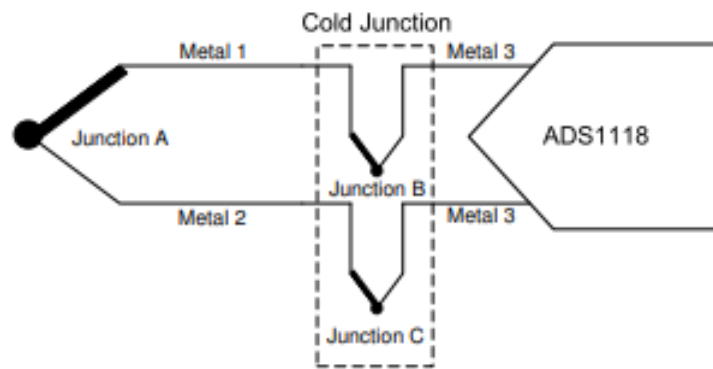


Figura 11 – Diagrama da junção do termopar ADS1118. Fonte: (BECKMAN; CHIOYE, 2011).

Conforme explicitado na norma internacional IEC 584-2, este tipo de termopar não deve ser utilizados em atmosferas redutoras (meios gasosos que tendem a desoxidar materiais com os quais entram em contato) ou sulfurosas (meios que contém enxofre), exceto casos em que utiliza-se a devida proteção. Recomenda-se este termopar para medidas de temperatura em ambientes oxidantes, visto que ele não sofre oxidação em atmosferas úmidas.

3.1.6 Indicador LCD:

O indicador *Liquid Crystal Display* (LCD) é um componente de fundamental importância em projetos aos quais necessita-se de visualização dos dados em tempo real, visto que ele pode facilitar a detecção e correção de erros durante as simulações e testes do *software*. O modelo do *display* integrado na placa *430BOOST-ADS1118* é o NHD-C0216CZ-FSW-FBW-3V3 *Serial Interface* e a conexão de seus pinos pode ser vista na Fig. (12). Ele possui duas linhas de 16 caracteres, onde decidiu-se usar uma linha para exibir a hora e a atualização da temperatura medida. Além disso, este LCD permite ao usuário o envio de mensagens, para que pessoas próximas ao instrumento leiam de imediato. Por exemplo, pode-se dizer: "Teste nº 1".

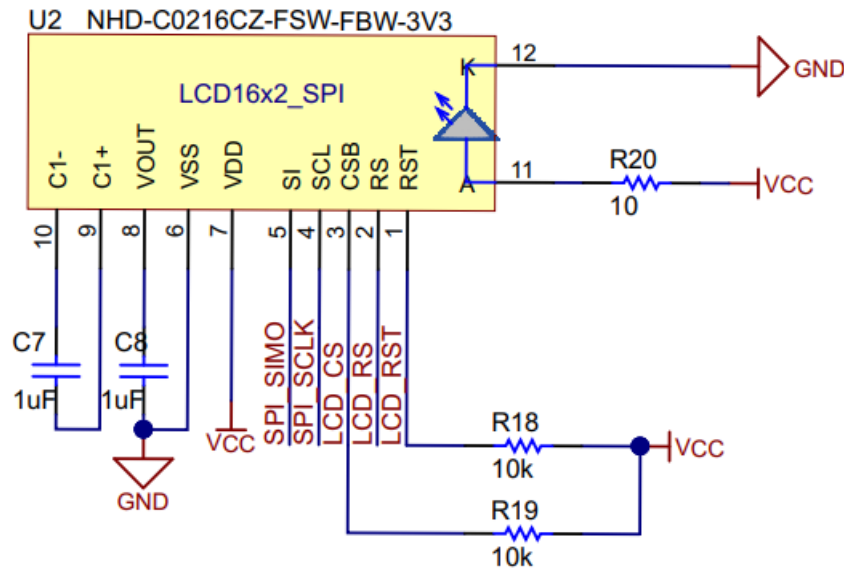


Figura 12 – Diagrama de pinos do circuito do display LCD. Fonte: (INSTRUMENTS, 2013).

3.1.7 Placa de adaptação

Tendo em vista que os ruídos são provenientes de fontes de emissão de ondas eletromagnéticas e seus campos associados, geralmente formados por “antenas acidentais”, foi necessário projetar caminhos que minimizam este problema.

Em substituição aos fios (*jumpers*) utilizados inicialmente para interligar os terminais dos pinos que fazem conexão entre duas placas empregadas, foi utilizada uma placa PCB universal com furos envoltos por condutores, onde as barras de pinos e o soquete foram inseridos e em seguida, soldados. A Figura (13) mostra os componentes utilizados nesta interface.

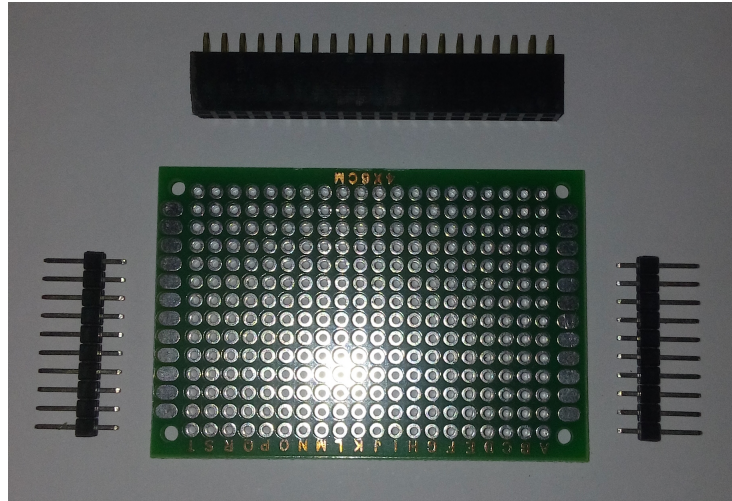


Figura 13 – Placa pré-furada, soquete e barras de pinos. Fonte: O autor.

3.2 CONEXÃO DE HARDWARE E INTERFACE

O protocolo de comunicação foi estabelecido e utilizando-se a placa de adaptação, realizou-se a conexão entre a RPi e o módulo ADS1118.

3.2.1 Protocolo de comunicação

A placa ADS1118 fez interface com a *Raspberry* utilizando-se um método de comunicação baseado em padrões SPI (*Serial Peripheral Interface*), por meio da conexão entre os pinos listados na Tab. (5):

Termopar ADS1118/Pin	RaspBerry/Pin/GPIO
P1.1 - VCC	1 - 3.3V - DC Power
P1.7 - CLK	23 - GPIO.11 - SPI CLK
P1.8 - ADS_CS	26 - GPIO.07 - SPI CE1
P2.8 - LCD_CS	24 - GPIO.08 - SPI CE0
P2.9 - LCD_RS	11 - GPIO.17 - GEN0
P2.6 - GND	9 - GND
P2.7 - SIMO	19 - GPIO.10 - SPI MOSI
P2.1 - SOMI	21 - GPIO.09 - SPI MISO

Tabela 5 – Pinos de conexão, interface ADS1118 - *Raspberry Pi Zero W*. Fonte: O autor.

- **Comunicação SPI:** Conta com quatro canais conhecidos como CS, SCLK, MISO e MOSI e uma atribuição de mestre e escravo para os dois dispositivos envolvidos na comunicação. No caso da placa RPi e o ADS1118, a RPi é o dispositivo principal, responsável por controlar o sinal de *clock*, transmitindo dados por meio do pino MOSI e recebendo dados através do pino MISO. Desta forma, a transferência de

dados pode ocorrer simultaneamente e de modo bidirecional. Quando os dados são unidirecionais, os sinais de MOSI ou MISO podem ser ignorados, considerando-se a direção de interesse.

Portanto, nesta Interface Periférica Serial (SPI), para enviar e receber dados, a RPi é o mestre e o ADS1118 o escravo. A Figura 14 mostra a conexão entre os barramentos do ADS1118 e da RPi. Nesta implementação foram usados os pinos de 1, 2, 3, 8, 9 e 10.

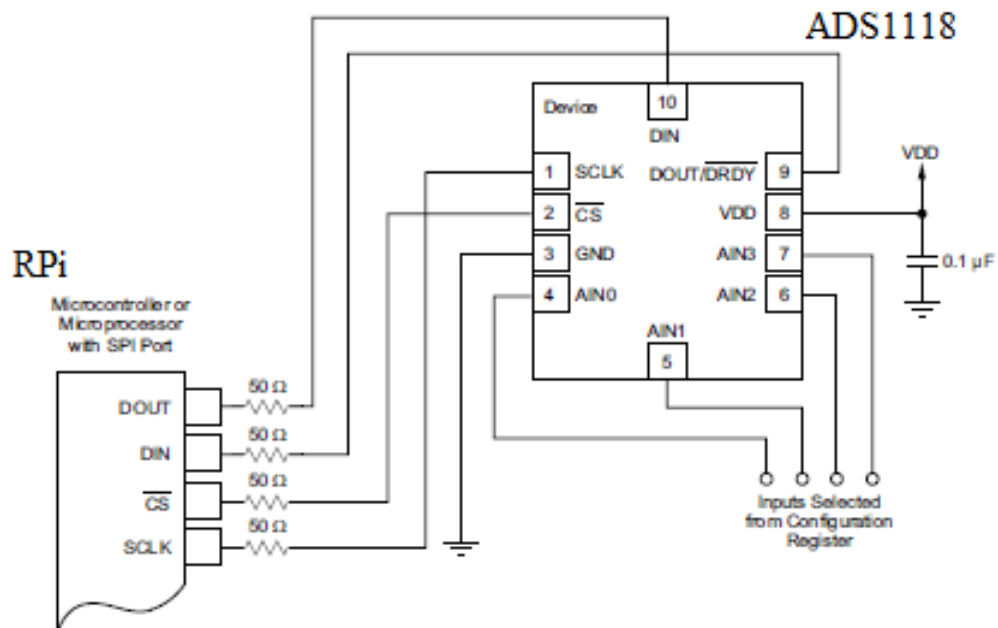


Figura 14 – Interface de *hardware* – barramento de conexão SPI, onde os canais SCLK, CS, DIN e DOUT foram utilizados para fazer interface de RPi com o *display*.
Fonte: BECKMAN; CHIOYE, 2011, adaptado.

O *datasheet* do ADC apresenta algumas informações sobre o ciclo de transmissão e os dados que precisam ser enviados ao ADS1118 para configurar os registros de controle. O ADS1118 possui dois modos de operação: 1. ciclo de transmissão de dados de 32 *bits* e 2. ciclo de transmissão de dados de 16 *bits*. A Figura 15 mostra o ciclo de transmissão de dados de 16 *bits*, ao qual foi o método adotado nesta implementação.

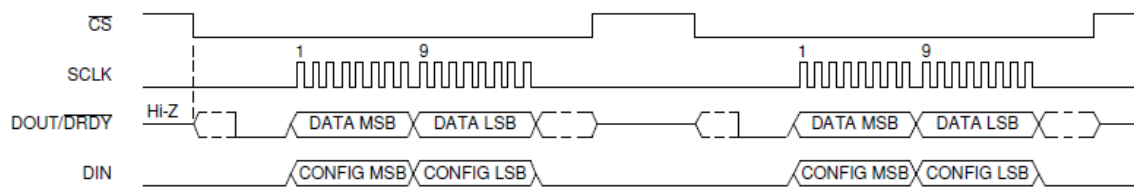


Figura 15 – Ciclo de transmissão de dados de 16 *bits*. Quando os dados do registro de configuração não precisam ser lidos, os dados de conversão ADS1118 podem ser sincronizados em um curto ciclo de transmissão de dados, de 16 *bits*. Quando o canal SCLK é mantido baixo por 28 *ms*, a interface serial é redefinida e o próximo pulso SCLK inicia um novo ciclo de comunicação. Fonte: (INSTRUMENTS, 2019a).

O canal CS, controlado pela RPi, encontra-se em nível baixo permitindo o envio de uma *string* de 16 *bits* para o ADS1118 na linha MOSI, esta *string* contém os ajustes de configuração do registro de controle. Como a interface SPI permite que os dados sejam transmitidos e recebidos simultaneamente, o ADS1118 envia uma *string* de 16 *bits* de volta na linha MISO. O mestre então assume o canal CS em nível alto, e outra transferência de *strings* de 16 *bits* pode ocorrer assim que ela for mudar de estado novamente.

Portanto, o canal CS é acionado após o décimo sexto ciclo de SCLK, sendo que quando CS é desativado ocorre uma redefinição na interface SPI e em seguida a transmissão de dados inicia com o resultado da conversão em um *buffer* na primeira borda ascendente do SCLK. Neste sistema, o nível do canal DOUT/DRDY fica baixo quando é iniciada a recuperação de dados, enquanto o *buffer* de conversão é atualizado com um novo resultado. Em contrapartida, quando o canal DOUT/DRDY encontra-se em nível alto, ocorre a leitura do mesmo resultado do ciclo de transmissão de dados anterior.

As configurações do registro de controle para o ADS1118 são usadas para configurar vários parâmetros, como o modo do sensor de temperatura, resistores *pull-up*, configuração do amplificador de ganho programável, entre outros. Nesta implementação, o ADS1118 foi configurado para funcionar em modo de detecção de temperatura, usando o sensor de temperatura interno, sendo que para configurar o registro de controle, uma *string* de 16 *bits* em base binária foi definida como 0000 0000 1111 1111 ou 0xFF em hexadecimal. O *bit* 4 foi definido como 1, ativando o modo de sensor de temperatura do dispositivo, sendo que quando os dados do registro de controle são classificados como 'dados válidos', os *bits* 1 e 2 têm configuração fixa, também em nível lógico alto. Quando isso não ocorre, significa que a *string* de 16 *bits* é inválida e, portanto, não pode ser gravada no registro de controle.

A velocidade máxima de amostragem do ADS1118 é de até 860 amostras por segundo. Entretanto, esta implementação foi configurada para capturar 8 amostras por segundo e, em seguida, calcular a média dos resultados uma vez por segundo. Essas

amostras puderam ser calculadas pela RPi produzindo resultado de conversão a cada 125 *ms*. Desta forma, outras configurações também podem ser implementadas, desde que seja considerado o limite de 860 amostras por segundo.

3.2.2 Conexão de hardware

A substituição de fios (*jumpers*) pela placa PCB universal proporcionou um caminho que reduziu efeitos indesejáveis na transmissão de dados, causados por mau contato e Emissão Eletromagnética (EMI), na região de conexão entre os terminais. A Figura (16) mostra a placa de adaptação sendo utilizada.



Figura 16 – Conexão sendo realizada. Fonte: O autor.

Para aperfeiçoar o protótipo em desenvolvimento, dentre outras soluções, é possível projetar um recipiente com blindagem específica, utilizado para cobrir o sistema e reduzir de forma eficaz a EMI, refletindo e absorvendo parte da sua radiação. A Figura (17) apresenta a conexão entre a *Raspberry Pi* e a *TI 430BOOST-ADS1118*, por meio da placa de adaptação.

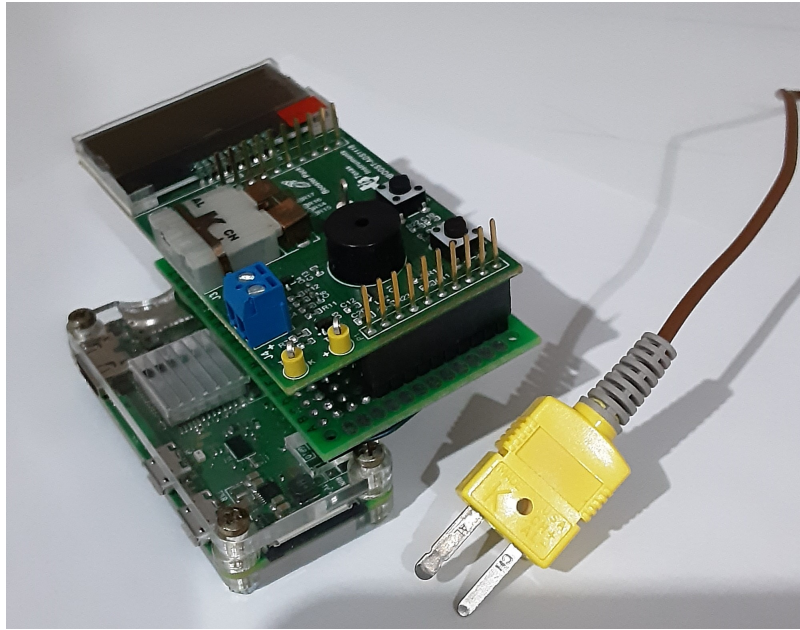


Figura 17 – Protótipo em desenvolvimento. Dispositivo de monitoramento de temperatura via *Wi-Fi*. Fonte: O autor.

3.3 DESCRIÇÃO DE SOFTWARE

Esta seção descreve todo o *software* utilizado. Primeiramente é apresentado como se deu o processo de instalação do sistema operacional e pacotes de suporte para acesso remoto à RPi. Em seguida, revela-se os principais comandos e *firmwares* utilizados para aferir a temperatura, além de diagramas que auxiliam na compreensão do sistema e a implementação de um servidor *web* que facilita o acesso e compartilhamento em uma página *web*.

3.3.1 Configuração - *Raspberry Pi* e *Linux*

Inicialmente, a instalação do *Raspbian* foi realizada utilizando-se um cartão microSD. Este sistema operacional é uma versão não oficial do *Wheezzy Debian Linux*, otimizada para aplicações em RPi. Ele conta com as principais características do *Debian*, sendo um *software* gratuito, fácil de ser instalado e que possibilita um suporte contínuo da comunidade, a qual disponibiliza atualizações e correções de *bugs*.

Para configurar os módulos deste projeto, foi necessário realizar uma comunicação entre a RPi e o computador pessoal utilizado para acessar o dispositivo. Como a comunicação sem fio foi considerada vantajosa, empregou-se a rede *Wi-Fi*, para gerenciar a RPi por linhas de comando, por meio do protocolo *Secure Shell* (SSH) no *software* de emulação de terminal *Putty*. Dessa forma, verificou-se que existe a possibilidade de acessar a área de trabalho do *Raspbian*, em uma janela aberta no *notebook*, usando o aplicativo *Virtual Network Computing* (VNC). Conforme apresentado na Fig. (18).



Figura 18 – Imagem da tela inicial do Sistema Operacional *Raspbian*. Fonte: O autor.

Essa forma eficiente de acesso permitiu escrever, editar e transferir os códigos para a RPi e executá-los remotamente usando um notebook. Nesta comunicação, verificou-se a facilidade de debugar códigos, a possibilidade de observar o processamento em tempo real e identificar eventuais falhas.

Durante os testes, utilizou-se o sistema operacional *Raspbian (Wheezy)* instalado com o *Noobs* versão 1.5.0 e atualizado posteriormente para a versão *Jessie*.

3.3.2 Medição de temperatura

Nessa etapa, um pseudocódigo escrito em C, disponibilizado pela *Texas Instruments*, foi adaptado para recuperar as medições de temperatura realizadas pelo sensor termopar conectado à RPi. Pode-se verificá-lo no Anexo B.

O início deste código define as bibliotecas e declara as variáveis globais utilizadas. Em seguida são definidas as funções globais. A seguir, uma descrição resumida das principais funções do código é apresentada:

- *delay_ms* usada para pausar por um curto período de tempo;
- *setup_io* usada para alocar memória;
- *lcd_writcom*, *lcd_writedata*, *lcd_clear*, *cd_display_string*, *lcd_init* são comandos para funcionalidade do LCD como leitura e escrita, inicialização, etc;
- *therm_transact* envia quatro bytes, dois *bytes* de configuração enviados duas vezes e retorna dois *bytes*;

- *local_compensation* transforma o código de temperatura interna para um código de compensação de temperatura que é adicionado ao código do sensor termopar;
- *adc_code2temp* converte os resultados do conversor AD para temperatura em uma faixa de -270 à 500 °C;
- *ads_config* configura e inicia a conversão;
- *ads_read* lê o resultado do conversor AD e inicia uma nova conversão;
- *get_measurement* retorna a temperatura medida;
- *get_measurement_fast* lê a medida do sensor de temperatura externo e reinicia o sensor externo.

Este algoritmo principal escrito em C, permite a comunicação entre o sensor de temperatura, a RPi e o *display* LCD, iniciando-se com a função *main()*. As primeiras linhas de código são definições de variáveis locais utilizadas no código C principal. Os canais de entrada e saída, GPIO, são os primeiros a serem chamados pelas funções *setup_io()*, *INP_GPIO()* e *OUT_GPIO()*.

Em seguida, a lógica para análise das entradas é realizada. A função *if (argc>2)* imprime mensagens no *display*. A função de lógica *if (argc>1)* inicializa o LCD e imprime o período decorrido entre cada medição de temperatura, além de uma mensagem pré-estabelecida pelo programador. Também existe uma lógica para encerrar o programa e imprimir uma mensagem de “finalização do processo” caso necessário. O laço *if (argc>3)* imprime no LCD as medidas de temperatura e uma mensagem de finalização.

A parte seguinte do código, permite usar a comunicação SPI com o termopar, fazendo-se uma medição de temperatura, imprimindo-a logo em seguida, e, retornando ao início do laço. Desta forma, as temperaturas podem ser obtidas a cada período de medição do termopar.

3.3.3 Detalhes do processo

As medições de tensão do termopar são realizadas usando a referência de tensão interna do ADS1118, enquanto as medições de compensação de junção fria (CJC) ocorrem por meio do sensor de temperatura integrado.

O procedimento de cálculo, para obter a compensação da junção fria, foi realizado intercalando leituras entre as entradas do termopar e a temperatura sensor. Ou seja, foi adquirido um resultado de temperatura no chip para cada tensão ADC do termopar medida.

Quando a junção fria fica em um ambiente estável, podem ser necessárias medições periódicas para o cálculo ser realizado adequadamente. Em cada medição de termopar e

temperatura de junção fria, as operações produzem dois resultados em um ciclo de medidas de junção: A tensão do termopar (V_{TC}) e a temperatura no chip ou (TCJC).

Para produzir a VCJC, o sensor do ADS1118 converte a variação de temperatura em uma tensão proporcional ao sinal do termopar tipo K, usado com a junção fria.

A adição das duas tensões gera a compensação na medida de tensão V_{Atual} , onde a “Eq. (3.1)” correspondente é:

$$V_{Atual} = VCJC + V_{TC} \quad (3.1)$$

V_{Atual} é então convertida para temperatura de acordo com os valores consultados na tabela da ITS-90 (NIST, 2014), resultando em T_{Atual} . Um diagrama de blocos mostrando esse processo é apresentado na Fig. (19).

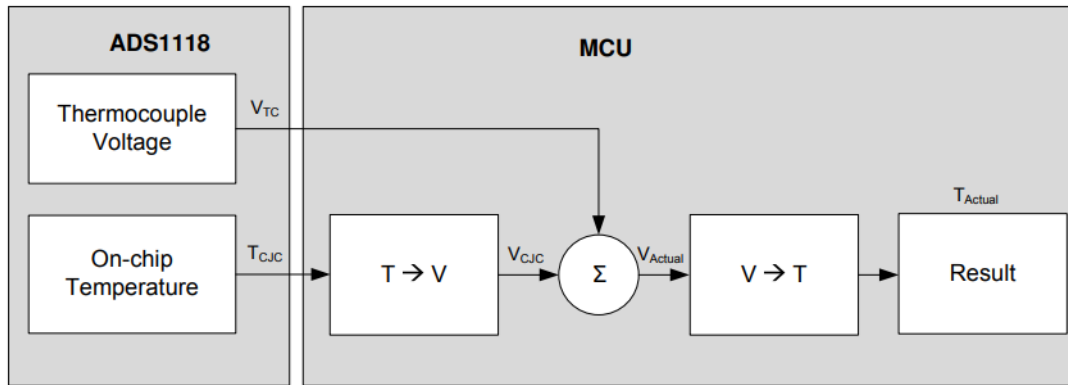


Figura 19 – Diagrama de blocos - Fluxo do *software*. Fonte: BECKMAN; CHIOYE, 2011.

Tanto a conversão de temperatura em tensão quanto a conversão de tensão em temperatura podem ser realizadas utilizando coeficientes que podem ser representados por um polinômio de ordem elevada, a qual é programado no microcontrolador (MCU) para apresentar os respectivos resultados em cada leitura. Embora este método ofereça o menor erro introduzido durante a conversão, é extremamente intensivo para o processador e não é prático para maioria das aplicações. Por este motivo, optou-se por empregar o método de conversão por meio do uso de tabelas (Fig. 20) recomendadas pelo fabricante.

Este método oferece uma excelente precisão para linearizar as medidas, de aproximadamente 1°C para cada valor. Tendo em vista que o número de entradas usadas na tabela afeta a precisão da conversão, geralmente, utiliza-se de 16 a 32 entradas da tabela de pesquisa. Além disso, elas não precisam ser igualmente espaçadas.

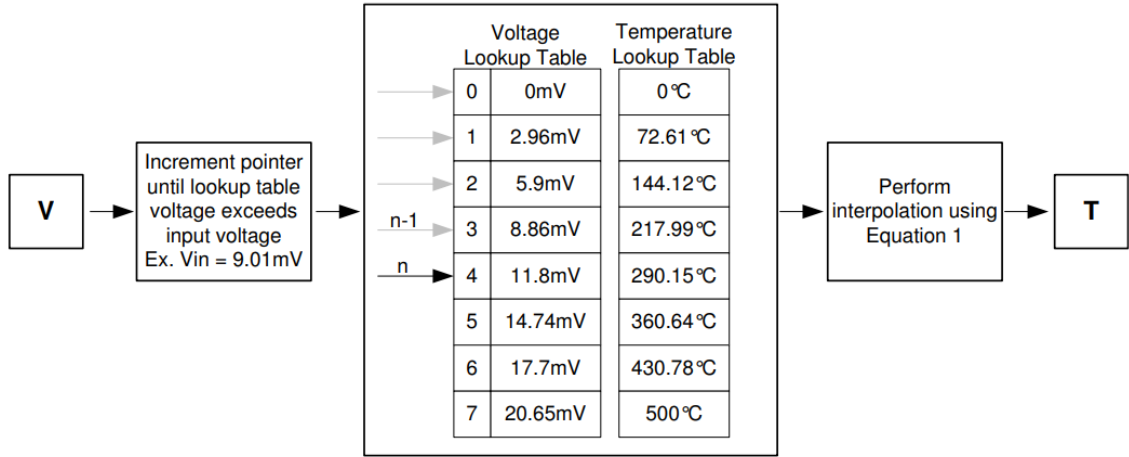


Figura 20 – Exemplo do método ao converter de tensão (V) em temperatura (T), usando tabelas de consulta. Fonte: BECKMAN; CHIOYE, 2011.

Para executar uma interpolação linear usando uma tabela de pesquisa, o valor convertido foi comparado com os valores da tabela, até exceder o valor que está sendo convertido. Desta forma, a “Eq. (3.2)” foi usada para converter tensão para temperatura, em que VLT é a matriz da tabela de tensão e TLT é a matriz da tabela de temperatura.

Esta operação envolve quatro adições, uma multiplicação e uma divisão, respectivamente. E, pode ser realizada facilmente na maioria dos microcontroladores de 16 e 32 *bits*.

$$T = TLT[n - 1] + (TLT[n] - TLT[n - 1]) \left(\frac{VIN - VLT[n - 1]}{VLT[n] - VLT[n - 1]} \right) \quad (3.2)$$

Conforme mostrado na “Eq. (3.3)” a conversão de temperatura para tensão é semelhante.

$$V = VLT[n - 1] + (VLT[n] - VLT[n - 1]) \left(\frac{TIN - TLT[n - 1]}{TLT[n] - TLT[n - 1]} \right) \quad (3.3)$$

Levando em consideração que o coeficiente de verificação do termopar Tipo K é de aproximadamente $40 [\mu V/^{\circ}C]$ na faixa de 0 a $1000 [^{\circ}C]$, foi utilizada uma faixa de entrada no conversor analógico digital de $\pm 256\text{mV}$. Esta configuração foi realizada por meio de *strings* de 16 *bits*, onde o valor do *bit* menos significativo (LSB) para o circuito ADC foi de $7,8125[\mu V]$. Este valor foi usado para converter o valor digital no ADC para uma tensão medida e vice-versa. A ficha técnica do termopar indica a tensão versus temperatura onde o ADC pode ser programado para produzir uma contagem para cada sinal de $7,8125 [\mu V]$, com a finalidade de determinar a temperatura em que a conversão é necessária. Neste cenário, verificou-se que devido à linearidade do termopar, diferentes coeficientes

podem ser usados em faixas menores de temperatura. Em relação à calibração, existem algumas fontes de erro inerentes ao ADC, para descobrir o impacto real no valor medido, segundo a folha de dados do ADS1118, seria necessário analisar parâmetros como o erro de deslocamento de *bits*, erro de ganho e qualquer não linearidade.

3.3.4 Levantamento de pontos e possibilidade de realizar calibração

Levando-se em consideração que as propriedades do par de condutores podem variar conforme o uso em processos de aferição de temperatura, no decorrer do tempo, é importante garantir que o dispositivo termopar mede dentro dos limites esperados, fazendo-se necessário calibrar o insumento.

De acordo com o INMETRO ([INMETRO, 2020](#)), uma das formas de calibração aceitas é a calibração por comparação, utilizando-se um termopar calibrado como base de referência durante o ensaio de calibração, além da consulta de materiais disponibilizados pela ITS90.

Para verificar o desempenho do sistema implementado, utilizando o ADS1118, foi realizada um levantamento de medidas pontuais, com a finalidade de comparar suas indicações com as respectivas medidas do dispositivo de referência TM-902C. Este dispositivo de referência pode ser visto na Fig. (21).



Figura 21 – Dispositivo TM-902C com termopar tipo K. Instrumento medidor de temperatura usado durante o ensaio de medidas de temperaturas. Fonte: O autor.

O fogareiro mostrado na Figura 22 foi utilizado durante o processo de levantamento de medidas pontuais. Este dispositivo aquecedor permite a transformação de energia elétrica em energia térmica, a qual é dissipada através de uma resistência elétrica que aquece proporcionalmente a intensidade de corrente elétrica aplicada.



Figura 22 – Fogareiro (WY-01B) utilizado durante o ensaio de levantamento de medidas pontuais, com a finalidade de transmitir calor por meio de contato, como uma fonte variável de temperatura. Fonte: O autor.

O ensaio foi realizado por meio da inserção das pontas de prova, de ambos os termopares, tipo K, na superfície de metal aquecido do fogareiro elétrico, a qual teve sua temperatura aumentada gradualmente, cobrindo na faixa de temperatura de 25 a 505 °C, ajustando-se a intensidade de corrente elétrica por meio de um potenciômetro, conforme apresentado na Fig. (23). Desta forma, a temperatura de superfície foi ajustada para cada um dos pontos de teste, com um intervalo de 15 °C. Para comparar a leitura de temperatura medida por meio do 430BOOST-ADS1118 com a leitura realizada pelo TM-902C, os valores obtidos por estes instrumentos foram registradas em planilhas, conforme mostrado na seção 4.1.

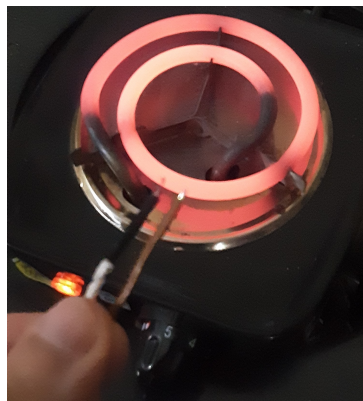


Figura 23 – Ensaio de levantamento de medidas pontuais. Medindo a temperatura em pontos fixos, na superfície aquecida do fogareiro elétrico, por meio dos dispositivos termopares, com a finalidade de comparar os valores obtidos e fazer ajustes contando com o auxílio de tabela de conversão de tensão/temperatura para termopar tipo K. Fonte: O autor.

A temperatura ambiente, medida durante a realização do ensaio, foi de aproximadamente 25 °C. A variação da temperatura em condições ambientais justifica o uso de valores de compensação de junção fria atribuídos ao *software* do dispositivo termopar, tais valores podem ser consultados em tabelas de referência, levando-se em conta as especificações do termopar utilizado, conforme a ANSI MC 96.1.

A tabela de conversão da ITS-90 (NIST, 2014) foi consultada durante a leitura e análise dos valores de temperatura, com a finalidade de converter os valores de tensão, em milivolts, para valores equivalentes de temperatura, em graus Celsius, utilizando o método da interpolação para valores medidos entre dois pontos de calibração consecutivos, conforme apresentado na seção 3.3.3.

Verificou-se que existe a possibilidade de analisar os valores em intervalos menores de temperatura e encontrar o *offset* viabilizando-se o fornecimento dos respectivos valores de ajuste para compensação de erro sistemático. Esta correção pode ser realizada obtendo-se a diferença entre o valor de referência e a média das leituras obtidas pelo dispositivo. Contudo, esta correção não foi realizado neste projeto, ficando como uma proposta para trabalhos futuros.

3.3.5 Servidor Web

Com o intuito de acompanhar as medidas remotamente, via *Wi-Fi*, foi realizada a implementação de uma página de acesso por meio de navegador *web*. Esta interface se deu por meio de um *framework web* baseado em processos cujas funções foram adaptadas para permitir a operação do servidor (*back-end*) e também, em processos da camada do usuário, para simplificar a leitura e navegação no *browser* (*front-end*).

Durante a implementação da interface gráfica de acesso por meio do *browser*, utilizou-se, como referência, modelos escritos em *JavaScript* e Linguagem de Marcação de Hipertexto (HTML) formatada por meio de folhas de estilo em cascata (CCS), como a aplicação criada por Yousaf (YOUSAF, 2014), a qual inclui o módulo *Node.js*, um *software open source* que permite a execução de códigos em *JavaScript* a nível *front-end* e *back-end*.

Por meio da biblioteca *Socket.io* foi possível realizar a comunicação bidirecional, a qual foi condicionada por instruções que permitem a interação entre o servidor *web* e navegador *web*, viabilizando a comunicação entre cliente e servidor e a atualização constante dos valores medidos de temperatura, aproximadamente em tempo real.

Utilizando o terminal do SO *Raspbian*, foram instalados os pacotes dos módulos *Node.js* e *Socket.io*, por meio do comando "*npm install socket.io*". Desta forma, o acesso *web* pode ser representado pelo diagrama presente na Fig.(24):

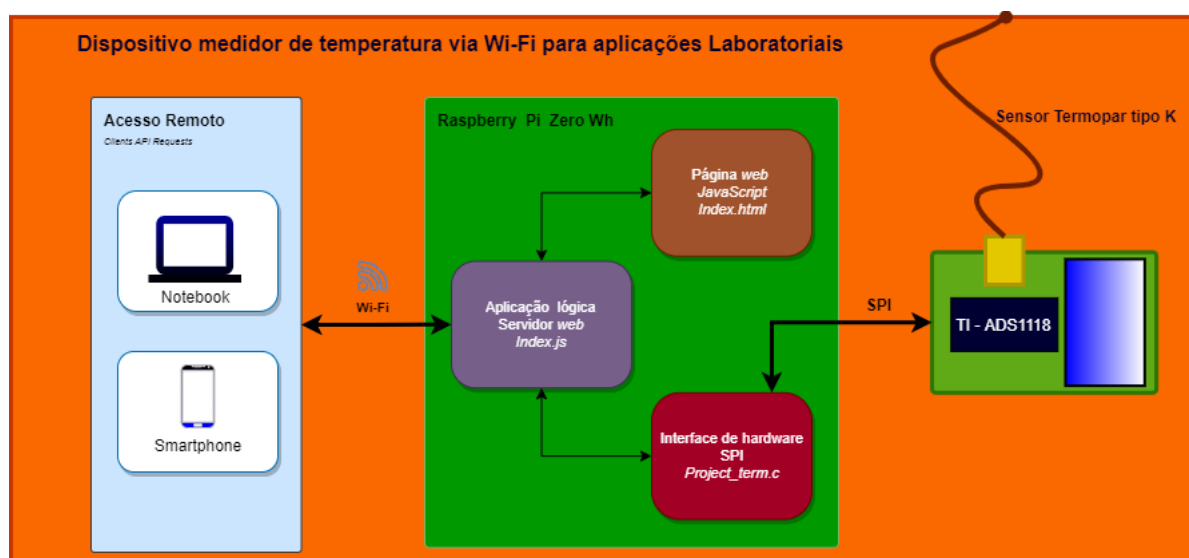


Figura 24 – Diagrama geral da arquitetura do dispositivo medidor de temperatura. Fonte: O autor.

Ao executar o *script* de interface de *hardware*, inicia-se a leitura por meio do sensor e o servidor *web* é executado na porta 8081 da RPi, ao qual responde as requisições HTTP de envio e exibição de dados de temperatura e de tempo na página *web*. A Figura 25 apresenta um fluxograma utilizado nesta implementação.

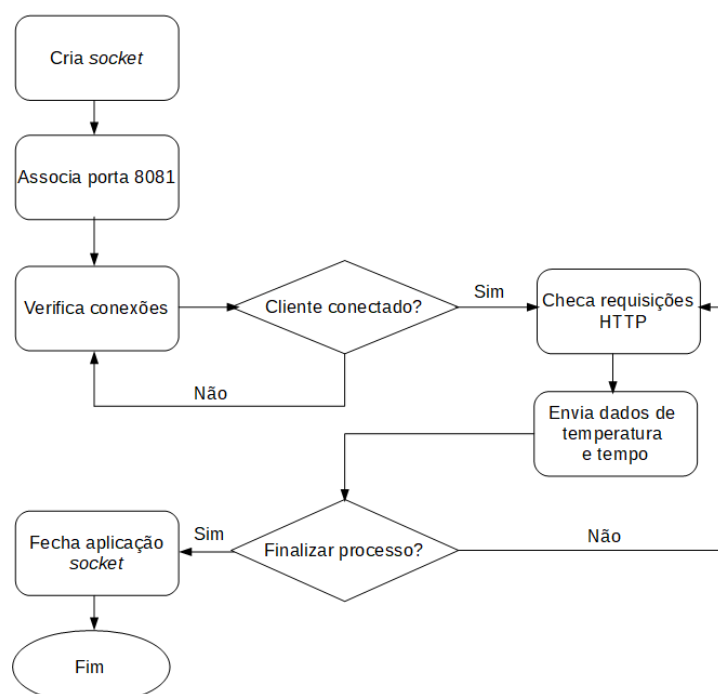


Figura 25 – Fluxograma simplificado de comunicação utilizando *socket* para criar uma interface *web*. Fonte: O autor.

A forma como as informações são transmitidas ao servidor se dá por meio do

recurso do *Socket.IO*, em especial o *socket.emit*, uma ferramenta conveniente para lidar com transmissão de parâmetros (eventos e dados), permitindo o envio de *strings* e *arrays* para o servidor codificado em *NodeJS*. Nesta implementação (Anexo B), para realizar a transmissão dos dados de interesse, foi usada a função *split()* a qual divide uma *String* em uma lista ordenada de *substrings*, coloca essas *substrings* em um *array* e retorna o *array* ao destino especificado. A título de exemplo, na Fig.(26) pode-se revelar alguns trachos onde utilizou-se esta função:

```

1  #!/usr/bin/env node // instância shebang Unix
2
3  // Adicionando um servidor http
4  var app = require('http').createServer(handler)
5  , io = require('socket.io').listen(app)
6  , fs = require('fs');
7  var path = require('path');
8
9  //iniciando a aplicação usando a porta 8081
10 app.listen(8081);
11
12 // Variaveis
13 var child=require('child_process');
14 var prog;
15 var prog2;
16 var runscope=0;
17 var progpath="/home/pi/development/therm/";
18 var values = "abc";
19 var ifile = progpath;
20 var child2=require('child_process').exec;
21 var cprog2;
22 var oldinfo1='1';
23 var oldinfo2='2';
24
25 // HTML --- Função de Middleware
26 // Para acesso ao objeto de solicitação (req) e ao objeto de resposta (res),
27 function handler (req, res)
28 {
29   console.log('url is '+req.url.substr(1));
30   reqfile=req.url.substr(1);
31   // {
32   //   reqfile="index.html"; // permissão de acesso
33   // }
34   fs.readFile(progpath+reqfile,
35   // O parâmetro err recebe um objeto de erro se houver um erro, caso contrário, dados.
36   function (err, data)
37   {
38     if (err)
39     {
40       res.writeHead(505);
41       return res.end('Error loading index.html');
42     }
43
44     res.writeHead(200);
45     res.end(data);
46   });
47 }
48 // Resposta ao cliente - Escrevendo o cabeçalho de resposta na aplicação
49 function ads_adc(v)
50 {
51   prog=child.exec(progpath+'therm', function (error, data, stderr) {
52     values=data.toString();
53     console.log('old prog executed, values length is '+values.length);
54   });
55   prog.on('exit', function(code)
56   {
57     socket.emit('results', {measurement: values});
58     console.log('old app complete, values length is '+values.length);
59   });
60 }
61
62
63 // Gerenciamento de comunicações Socket.IO
64 io.sockets.on('connection', function (socket)
65 {
66   socket.emit('status', {stat: 'ready'});
67
68   socket.on('action', function (data)
69   {
70     var dummy="dummy cmd";
71     var temp=dummy.split(" ");
72     cmd=data.command;
73     console.log("command is "+cmd);
74     isgettemp=0;
75     isreadfile=0;
76     islogstart=0;
77     islogstop=0;
78     ischeckstate=0;
79     if (cmd=="gettemp")
80     {
81       isgettemp=1;
82     }
83     if (isgettemp)
84     {
85       //ads_adc();
86     }
87   }
88   else
89   {
90     cmd=cmd.toString(); // envio de dados ao cmd
91     temp = cmd.split(" ");
92     if (temp[0]=="logstart")
93     {
94       islogstart=1;
95     }
96     else if (temp[0]=="logstop")
97     {
98       islogstop=1;
99     }
100    else if (temp[0]=="readfile")
101    {
102      .....
103    }
104    if (ischeckstate)
105    {
106      var state_is_logging=0;
107      prog=child.exec('ps -ef | grep therm', function (error, data, stderr) {
108        var lines = data.toString().split('\n'); //envio de dados,
109        lines.forEach(function(line) { //matriz de substrings
110          if (line.indexOf(progpath+'therm')>0)
111          {
112            if (line.indexOf('msg')>0)
113            {
114              // ok, confident it is logging
115              state_is_logging=1;
116            }
117          }
118        }); // end of socket.on('action', function (data)
119      }); // end of io.sockets.on('connection', function (socket)

```

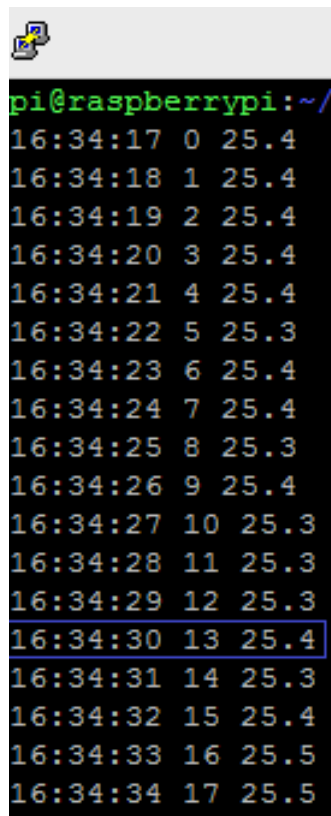
Figura 26 – Trechos do código em *Java Script*, implementado utilizando-se, dentre outras ferramentas, *String()* e *split()* para dividir *strings* em matrizes de *substrings* e retornar novas matrizes. Fonte: Yousef, adaptado.

4 RESULTADOS

Finalizada a conexão de *hardware* e a implementação da interface de *software* da plataforma, foram iniciados testes por meio do terminal do Sistema Operacional *Raspbian*.

4.1 Aferindo a temperatura

Foram realizados testes medindo-se a temperatura ambiente com o termopar. Esses valores foram atualizados e exibidos a cada segundo no LCD e simultaneamente, apresentados no terminal de linha de comando (*shell*). A Figura 27 apresenta os resultados impressos no *shell* enquanto a Figura 28 apresenta a amostra correspondente exibida no *display* de LCD.



A terminal window on a Raspberry Pi showing a list of temperature readings. Each line contains a timestamp in HH:MM:SS format, a counter from 0 to 17, and a temperature value in degrees Celsius. The line for 16:34:30 is highlighted with a blue box.

Timestamp	Counter	Temperature (°C)
16:34:17	0	25.4
16:34:18	1	25.4
16:34:19	2	25.4
16:34:20	3	25.4
16:34:21	4	25.4
16:34:22	5	25.3
16:34:23	6	25.4
16:34:24	7	25.4
16:34:25	8	25.3
16:34:26	9	25.4
16:34:27	10	25.3
16:34:28	11	25.3
16:34:29	12	25.3
16:34:30	13	25.4
16:34:31	14	25.3
16:34:32	15	25.4
16:34:33	16	25.5
16:34:34	17	25.5

Figura 27 – Temperaturas medidas, a cada segundo, e enviadas ao usuário que acessa o sistema remotamente. Fonte: O autor.

Pode-se observar na Figura 28 o horário de realização da medição, o número da medida correspondente e a temperatura obtida em °C.

Verificando-se as Figuras 27 e 28 observa-se que a 13ª medida, realizada às 16h:34m:30s, apresentou 25,4 °C, resultado a qual foi exibido no *display* de LCD.



Figura 28 – Resultado, em tempo real, da medida de temperatura e horário exibido. Fonte: O autor.

4.1.1 Desenvolvimento da página web

Nesta página *web*, além do bloco que exibe o valor da temperatura medida e o horário, foi implementado outro bloco onde o operador pode digitar o intervalo de tempo entre as medidas a qual deseja-se obter, o nome do arquivo “nome.csv” e selecionar a função de registro de medidas “log”, que possibilita o acionamento dos botões, aos quais permitem iniciar e finalizar o processo de salvamento automático em memória externa e também a opção de realizar o *download* de um arquivo com extensão “.csv” contendo os dados registrados separados por vírgulas. Estes dados podem ser organizados em colunas, automaticamente, ao importá-los e tratá-los por meio de alguma ferramenta de edição de planilhas, tal como o Excel.

Através da página *web*, foi possível acessar o sistema, e realizar a leitura por meio de um *notebook* conectado à rede. A Figura 29 mostra o resultado desta implementação.



Figura 29 – Compartilhamento e registro de medidas de temperatura via comunicação sem fio. Por meio da conexão *Wi-Fi*, as informações podem ser enviadas utilizando-se a arquitetura cliente-servidor a qual foi implementada para facilitar o acesso e salvamento dos dados, utilizando um navegador *web*. Fonte: O autor.

Conforme exposto na seção 3.3.5, esta página fica armazenada no servidor (RPi) e quando o operador “chama” a página, através de uma solicitação, este servidor, por meio de um navegador, permite a leitura do conteúdo HTML formatado utilizando-se CSS, e a leitura de conteúdos tais como textos e imagens que são apresentados ao operador.

Utilizando a função “log”, para fins de análise, foram registrados os valores medidos por meio do ADS1118. A Figura 30 mostra o botão da função selecionada.



Figura 30 – Escolhendo a função *log*, determinando um intervalo de medidas, um nome para o arquivo.csv e clicando em iniciar, pode-se registrar e salvar os valores lidos durante o ensaio de medidas de temperatura. Fonte: O autor.

Os valores medidos e registrados foram acompanhados por meio do terminal, conforme apresentado na Fig. (31)

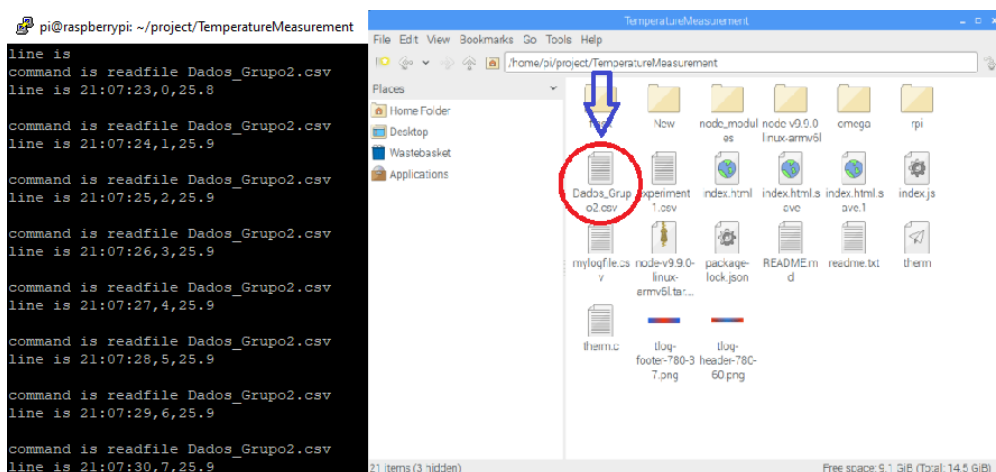


Figura 31 – Medidas sendo registradas em memória externa, por meio da função “log”. Acompanhamento do processo utilizando o terminal. Fonte: O autor.

4.1.2 Ensaio de medidas de temperatura utilizando uma resistência aquecida

Este ensaio foi realizado com o intuito de verificar a reprodutividade do método de medição utilizando o ADS1118. Durante os experimentos, as pontas de prova foram posicionadas sobre a resistência elétrica aquecida. A intensidade de temperatura foi ajustada, por meio de um potenciômetro, de acordo com os valores de temperatura desejados.

Os valores aferidos por meio do TM-902C foram anotados de forma manual, em uma planilha de dados, enquanto que os valores aferidos por meio do ADS1118 foram salvos automaticamente, por meio da função “log”. A Figura (32) apresenta a leitura, por meio do terminal, de alguns valores registrados e salvos utilizando-se a função “log”.



Figura 32 – Medidas sendo realizadas com o ADS1118 e salvas durante o ensaio de medição de temperatura. O período estabelecido altera a frequência de envio de dados de temperatura, isso implica diretamente no espaço de armazenamento destes em um determinado intervalo de tempo. Definindo 25 segundos como o período entre as medidas realizadas, os valores foram registrados e salvos por meio da função “log”. Fonte: O autor.

As temperaturas definidas por meio do dispositivo termopar TM-902C foram consideradas valores de referência para o estudo, pertencendo ao Grupo 1. Portanto, os valores de temperatura medidos por meio do sensor ADS1118, Grupo 2, estariam mais corretos quanto mais próximos ao valor de referência já conhecido. Com intervalos de temperatura

de 15°C, as medidas foram ajustadas numa faixa de 25° a 505 °C. Para cada uma destas temperaturas conhecidas, foram obtidas o mínimo de 3 amostras.

A Figura 33 , apresenta os valores medidos por meio do dispositivo termopar TM-902C.

Referência [°C]	Temperatura aferida usando TM-902C [°C]				
	Medida t1	Medida t2	Medida t3	Medida t4	Medida t5
25	25,1	25,3	24,9	24,8	25,2
40	40,0	40,2	40,5	39,9	40,4
55	55,3	55,1	55,2	55,1	54,9
70	69,9	69,8	70,3	69,9	70,2
85	84,5	84,8	85,2	85,0	85,1
100	100,3	100,2	101,4	100,7	99,7
115	115,8	115,3	114,6	115,5	115,2
130	130,2	130,6	128,6	129,3	129,7
145	145,6	144,7	144,8	145,4	145,3
160	159,4	158,9	159,7	160,1	161,2
175	175,3	175,1	174,9	174,7	175,6
190	191,2	190,5	189,6	191,7	189,3
205	205,4	205,9	204,7	203,9	205,6
220	220,8	221,3	220,5	219,6	219,2
235	235,9	233,7	234,9	236,3	236,1
250	250,7	248,2	248,9	253,2	251,5
265	263,3	262,9	264,2	267,1	266,4
280	282,1	280,6	281,9	277,2	282,8
295	295,7	293,9	294,8	291,1	295,4
310	309,9	306,3	312,7	310,7	305,9
325	323,6	328,3	321,9	324,4	326,5
340	341,9	342,3	335,1	343,0	339,2
355	354,2	351,9	354,6	352,1	349,9
370	372,5	368,5	369,7	373,2	370,9
385	385,3	383,4	383,8	385,3	385,5
400	399,9	400,2	396,8	395,9	399,0
415	411,7	414,0	418,7	418,3	417,8
430	435,6	423,9	434,1	425,7	426,9
445	439,9	450,4	447,3	446,7	439,0
460	460,4	459,7	457,0	462,9	451,1
475	475,6	473,9	474,8	475,3	475,2
490	493,8	492,1	486,3	489,3	495,8
505	499,9	500,4	497,2	496,9	498,5

Figura 33 – Medidas realizadas utilizando o TM-902C e registradas de forma manual, para cada valor de temperatura de referência, durante o ensaio. Grupo 1. Fonte: O autor.

Dessa forma, com a finalidade de eliminar amostras consideradas *outliers* e de garantir no mínimo três amostras, a tomada de medidas de temperatura, para cada valor de referência, foi repetida cinco vezes utilizando o ADS1118, conforme apresentado na Fig. (34).

Referência [°C]	Temperatura aferida usando ADS1118_RPi [°C]				
	Medida t1	Medida t2	Medida t3	Medida t4	Medida t5
25	22,6	28,9	23,7	24,9	27,3
40	45,3	42,5	37,9	44,1	43,4
55	56,8	55,7	52,8	52,5	56,9
70	73,7	64,2	76,3	73,2	67,1
85	82,5	78,4	83,6	83,3	84,1
100	106,4	98,2	92,7	105,9	103,3
115	118,1	110,9	119,4	116,7	112,9
130	126,9	125,4	135,5	133,6	127,2
145	142,5	148,9	143,4	147,8	146,3
160	155,3	166,5	164,8	165,7	164,0
175	172,2	173,5	177,4	175,8	174,1
190	197,4	197,7	189,3	196,4	195,3
205	208,0	201,4	195,8	204,1	202,6
220	224,8	215,3	217,3	218,4	221,9
235	230,1	238,7	238,0	236,9	233,2
250	243,3	249,5	241,9	258,1	259,2
265	269,6	258,4	264,2	268,3	263,7
280	286,5	288,9	273,7	269,2	282,9
295	295,1	297,6	282,8	299,4	289,2
310	296,4	299,3	308,1	310,9	314,8
325	325,9	323,8	327,0	321,3	328,4
340	350,2	338,5	347,9	336,7	435,8
355	361,5	352,4	349,2	352,1	354,5
370	362,4	367,0	376,8	372,7	361,9
385	379,1	382,2	381,9	385,3	387,3
400	406,9	391,8	389,6	393,2	398,9
415	410,5	402,9	422,4	412,7	399,3
430	425,6	438,8	434,6	433,1	439,6
445	448,3	439,1	447,9	438,4	455,3
460	466,5	456,4	443,2	461,0	452,7
475	469,8	457,6	472,8	493,2	499,2
490	498,6	494,9	487,8	484,3	509,8
505	496,4	476,3	488,1	518,4	514,2

Figura 34 – Medidas obtidas para cada valor de temperatura de referência, registradas automaticamente durante o ensaio, por meio do ADS1118 em conjunto com o sistema implementado. Grupo 2. Fonte: O autor.

5 ANÁLISE DOS RESULTADOS

Diferentemente de microcontroladores simples, que executam um programa de cada vez, a RPi é um computador de placa única (SBC – *single-board computer*), que pode executar múltiplas tarefas quase de forma simultânea. Seu processador e periféricos, permitem realizar interface com outras placas para trabalhar de forma conjunta, gerenciando e priorizando os processos estabelecidos. Os resultados experimentais das medidas de temperatura, usando o ADS1118, foram possíveis devido a interface baseado em um fluxo de *software* recomendado, utilizando-se tabelas de referência para estimar medidas através da interpolação linear.

Como existem muitas fontes de erro, verificou-se a dificuldade que se tem em alcançar alta precisão nas medidas com este sistema, levando-se em consideração que o termopar opera com base na Teoria Fundamental do Efeito Termoelétrico ou Efeito Seebeck, a qual diz que quando um condutor experimenta um gradiente de temperatura de uma extremidade do condutor para o outro, surge um potencial de tensão. Esse potencial de tensão surge porque os elétrons dentro do condutor se difundem em diferentes taxas, dependendo da temperatura. Desta forma, elétrons com maior energia no lado quente do condutor difundem-se mais rápido que os elétrons de baixa energia no lado frio. Isso resulta em um acúmulo de carga em uma extremidade do condutor, criando uma tensão potencial entre as extremidades quente e fria. Este efeito é ilustrado na Fig. (35).

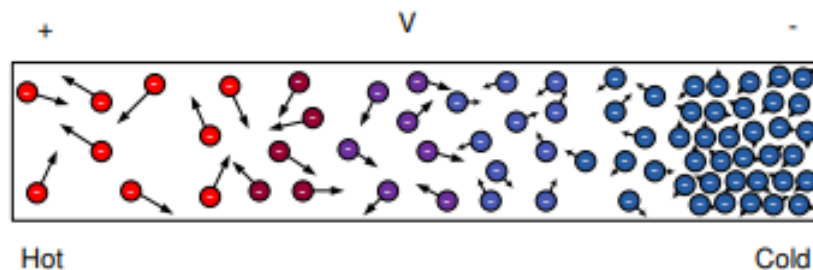


Figura 35 – Ilustração do Efeito Termoelétrico ou Efeito Seebeck. Fonte: BECKMAN; CHIOYE, 2011.

5.1 Interface web

Essa aplicação *web* pode facilitar a realização de experimentos científicos de longa duração, que requer a medida da variação de temperatura, por meio do termopar tipo K, ao qual cobre uma ampla faixa de temperatura através do contato com diferentes tipos de materiais.

Em relação a proposta de monitorar a temperatura em tempo real, por meio da rede *Wi-Fi*, os comandos que iniciam as funções lógicas, aqui tratadas, foram implementados em conjunto com o desenvolvimento de uma interface *web* utilizando HTML e *Java Script*.

Neste contexto, verificou-se que as funções escritas em *Java Script* podem ser embutidas dentro do *script* em HTML, permitindo o incremento das funcionalidades com diversos elementos que tornam possível responder aos eventos iniciados pelo usuário e incluir efeitos que tornam a página dinâmica.

A camada de comunicação entre navegador e o servidor *web*, usando *socket.io* permitiu o envio e recebimento de dados à interface gráfica e à *Raspberry Pi* com uma taxa de transmissão de dados de aproximadamente 1024 *kbps*. Esta interface gráfica dispõe de funcionalidades interativas, como a transmissão *web*, o uso botões virtuais e *download* de arquivo “.csv” contendo os dados de medidas de temperatura. Verificou-se que, nessa transmissão, podem ocorrer atrasos que são influenciados pela qualidade e velocidade da conexão *Wi-Fi* entre o usuário e o servidor.

Os dados registrados são iguais aos exibidos no *display*, no entanto, a escrita dos dados no arquivo “.csv” não ocorre ao mesmo tempo das medidas de temperatura exibidas no *display* e na página *web*. Isso acontece devido ao tempo de atraso durante a transmissão (aproximadamente 1,5s).

5.2 Análise de dados

De acordo com estudos realizados por Lerch, Nathal e Keller ([LERCH; NATHAL; KELLER, 2002](#)), a Estatística Aplicada permite verificar se existe um padrão no conjunto de medidas de temperatura, as quais foram realizadas variando-se a intensidade em tempos e/ou posições diferentes. Dessa forma, realizou-se a análise de dados, com a finalidade de encontrar divergências entre os grupos de medidas comparadas.

5.2.1 Eliminação de *outliers*

Alguns valores discrepantes foram detectados - por meio da função *isoutlier* utilizada no *software* MATLAB, conforme apresentado no Anexo C - e substituídos por valores aferidos além do mínimo de amostras estipulado. Tais discrepâncias podem ser devidas as condições físicas do ambiente, ao processo de ajuste manual de temperatura e à medições

imprecisas. Após substituir alguns valores considerados *outliers* (valores discrepantes), o banco de dados passou a ser analisado por meio do *software* IBM SPSS.

E com a finalidade de complementar a compreensão e análise, foram gerados gráficos de dispersão, aos quais apresentam o intervalo de confiança para cada valor médio das respectivas medidas de referência, para cada grupo. Estes gráficos podem ser vistos nas Fig.(36 e 37).

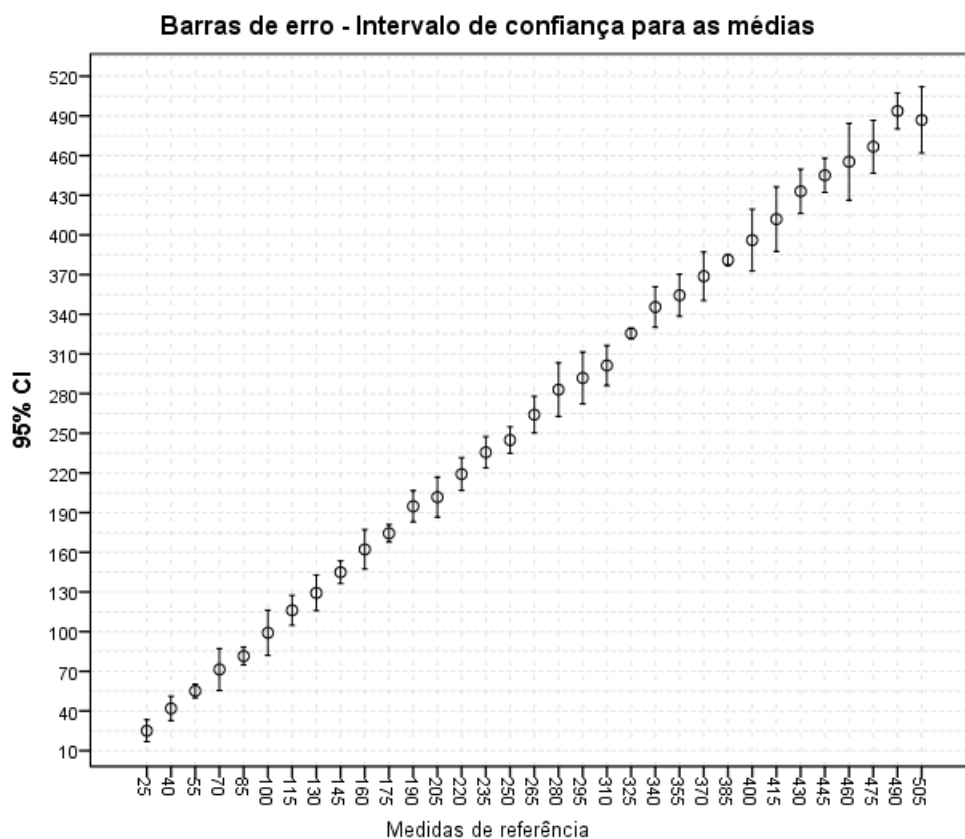


Figura 36 – Gráfico de dispersão com barras de erro das médias dos valores medidos entre 25 °C e 505 °C, considerando o intervalo de confiança de 95%. - Via ADS1118 (Grupo 2) – IBM SPSS. Fonte: O autor.

Analisando os gráficos de dispersão, pode-se notar que embora exista uma correlação entre as medidas pontuais, é difícil de se comparar e identificar quais as diferenças entre os grupos, fazendo-se necessário realizar testes de análise de variância. Portanto, como trata-se de variáveis numéricas estatisticamente dependentes, medidas por meio de dispositivos distintos, para comparar os valores resultantes das medidas pontuais de temperatura na faixa estabelecida, foi realizada a Análise de Variância de Medidas Repetidas (ANOVA) de duas vias.

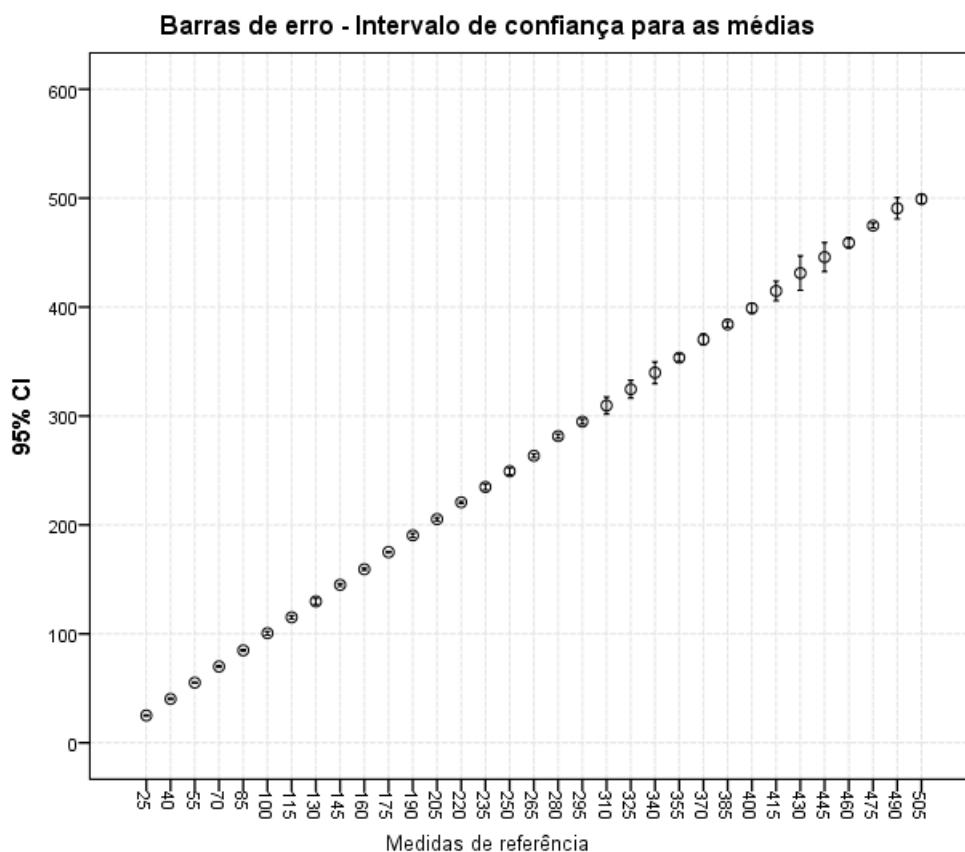


Figura 37 – Gráfico de dispersão com barras de erro das médias dos valores medidos entre 25 °C e 505 °C, considerando o intervalo de confiança de 95%. - Via TM-902C (Grupo 1) – IBM SPSS. Fonte: O autor.

5.2.2 Análise de variância de medidas repetidas de duas vias (ANOVA)

Considerando-se as variáveis independentes, fatores que variam para cada medida de temperatura de referência, sendo o tempo (medidas foram realizadas em instantes diferentes) e o tipo de medidor (ADS1118 ou TM-902C), é possível avaliar a influência sob as medidas empregando-se o método de análise supracitado, em um banco de dados que contém medidas de 33 valores fixos de temperatura, variando de 25 °C a 505 °C, conforme a Fig.(38). Cada linha está vinculada ao mesmo valor de temperatura de referência, estes grupos de valores tratam-se de medidas que são consideradas variáveis dependentes.

Referência [°C]	M [t1] - TM-902C	M [t1] - ADS1118	M [t2] - TM-902C	M [t2] - ADS1118	M [t3] - TM-902C	M [t3] - ADS1118
25	25,1	22,6	25,3	28,9	24,9	23,7
40	40,0	45,3	40,2	42,5	40,5	37,9
55	55,3	56,8	55,1	55,7	55,2	52,8
70	69,9	73,7	69,8	64,2	70,3	76,3
85	84,5	82,5	84,8	78,4	85,2	83,6
100	100,3	106,4	100,2	98,2	101,4	92,7
115	115,8	118,1	115,3	110,9	114,6	119,4
130	130,2	126,9	130,6	125,4	128,6	135,5
145	145,6	142,5	144,7	148,9	144,8	143,4
160	159,4	155,3	158,9	166,5	159,7	164,8
175	175,3	172,2	175,1	173,5	174,9	177,4
190	191,2	197,4	190,5	197,7	189,6	189,3
205	205,4	208,0	205,9	201,4	204,7	195,8
220	220,8	224,8	221,3	215,3	220,5	217,3
235	235,9	230,1	233,7	238,7	234,9	238,0
250	250,7	243,3	248,2	249,5	248,9	241,9
265	263,3	269,6	262,9	258,4	264,2	264,2
280	282,1	286,5	280,6	288,9	281,9	273,7
295	295,7	295,1	293,9	297,6	294,8	282,8
310	309,9	296,4	306,3	299,3	312,7	308,1
325	323,6	325,9	328,3	323,8	321,9	327,0
340	341,9	350,2	342,3	338,5	335,1	347,9
355	354,2	361,5	351,9	352,4	354,6	349,2
370	372,5	362,4	368,5	367,0	369,7	376,8
385	485,3	379,1	483,4	382,2	483,8	381,9
400	399,9	406,9	400,2	391,8	396,8	389,6
415	411,7	410,5	414,0	402,9	418,7	422,4
430	435,6	425,6	423,9	438,8	434,1	434,6
445	439,9	448,3	450,4	439,1	447,3	447,9
460	460,4	466,5	459,7	456,4	457,0	443,2
475	475,6	469,8	473,9	457,6	474,8	472,8
490	493,8	498,6	492,1	494,9	486,3	487,8
505	499,9	496,4	500,4	476,3	497,2	488,1

Figura 38 – Banco de dados com medidas repetidas. Onde TM-902C e ADS1118 são os dispositivos comparados e M[t1] refere-se as medidas, realizadas durante o ensaio, no intervalo de tempo 1, M[t2] no intervalo de tempo 2 e M[t3] durante o intervalo de tempo 3. Fonte: O autor.

Para realizar esse teste, foi necessário conferir se os dados possuem distribuição normal e/ou homogênea.

5.2.2.1 ANOVA e o Teste de Normalidade

A amostragem foi obtida por meio de três medidas de temperatura realizadas para cada valor de referência fixado. Para verificar se essa série de dados possui distribuição normal, aplicou-se o teste *Shapiro-Wilk*, onde foram dadas duas hipóteses:

Hipótese nula: Se $p < 0,05$ | H0: A distribuição é normal

Hipótese alternativa: Se $p > 0,05$ | H1: A distribuição não é normal

A Figura (39) apresenta a tabela com o resultado deste teste.

Tests of Normality						
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
M[t1]-TM-902C	,085	33	,200 [*]	,949	33	,125
M[t1]-ADS1118	,085	33	,200 [*]	,955	33	,183
M[t2]-TM-902C	,087	33	,200 [*]	,949	33	,126
M[t2]-ADS1118	,075	33	,200 [*]	,952	33	,154
M[t3]-TM-902C	,089	33	,200 [*]	,947	33	,109
M[t3]-ADS1118	,088	33	,200 [*]	,952	33	,148
*. This is a lower bound of the true significance.						
a. Lilliefors Significance Correction						

Figura 39 – Teste de Normalidade *Shapiro-Wilk*, considerando o nível de significância de $p=0,5$ e o coeficiente de confiança de 95,5%. Observa-se que $\text{Sig.} > 0,05$. Fonte: O autor.

Analisando-se o resultado do teste de normalidade para os conjuntos de medidas obtidos para cada medidor de temperatura, conforme a Fig.(39), como $p > 0,05$, o teste de *Shapiro-Wilk* mostra que a distribuição não é normal.

Portanto, como os dados não apresentam distribuição normal, aplicou-se a propriedade de logaritmo nestes valores, tendo em vista que o logaritmo dos dados torna a distribuição normal, conforme pode ser visto na tabela do teste após este procedimento (Fig.40), com valores de nível de significancia inferiores à 5%, viabilizando a realização de testes paramétricos, como a ANOVA.

Tests of Normality						
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
LogMt1TM902C	,138	33	,111	,893	33	,004
LogMt1ADS1118	,135	33	,136	,892	33	,003
LogMt2TM902C	,137	33	,117	,894	33	,004
LogMt2ADS1118	,144	33	,080	,890	33	,003
LogMt3TM902C	,139	33	,105	,892	33	,003
LogMt3ADS1118	,140	33	,098	,886	33	,002
a. Lilliefors Significance Correction						

Figura 40 – Teste de Normalidade *Shapiro-Wilk* após o tratamento de dados com função Logarítmica, considerando o nível de significância de $p=0,5$ e o coeficiente de confiança de 95,5%. Observa-se que $\text{Sig.} < 0,05$. Fonte: O autor.

5.2.2.2 ANOVA e o Teste de Esfericidade

Como foi obtida uma distribuição normal, o modelo estatístico ANOVA foi utilizado para fazer uma análise de variação intra e inter conjuntos de medidas de temperatura, ou seja, levando-se em consideração:

1. A variação dentro dos grupos: avaliando a discrepância entre as três medidas realizadas para cada valor de temperatura de referência/tipo de medidor.

2. E a variação entre grupos: avaliando a discrepância existente entre as medidas realizadas para diferentes grupos de temperaturas conhecidas.

Nesta análise de variância de medidas repetidas, convencionou-se um intervalo de confiança de 95% e nível de significância estatística de 5%.

O teste de esfericidade ou teste de *Mauchly's* tem como hipóteses:

H0: Existe esfericidade | $p > 0,05$

H1: Não existe esfericidade | $p < 0,05$

Conforme a Figura (41), a correção usada foi a de *Greenhouse-Geisser*, apresentando um valor de significância maior que 0,05, tanto para variável tempo quanto para a interação entre tempo e tipo de medidor, revelando que existe esfericidade.

Mauchly's Test of Sphericity ^a							
Measure: MEASURE_1							
Within Subjects Effect	Mauchly's W	Approx. Chi-Square	df	Sig.	Epsilon ^b		
					Greenhouse-Geisser	Huynh-Feldt	Lower-bound
Tempo	,966	1,063	2	,588	,967	1,000	,500
Tipo_de_Medidor	1,000	0,000	0		1,000	1,000	1,000
Tempo * Tipo_de_Medidor	,951	1,555	2	,460	,953	1,000	,500

Tests the null hypothesis that the error covariance matrix of the orthonormalized transformed dependent variables is proportional to an identity matrix.

a. Design: Intercept
Within Subjects Design: Tempo + Tipo_de_Medidor + Tempo * Tipo_de_Medidor

b. May be used to adjust the degrees of freedom for the averaged tests of significance. Corrected tests are displayed in the Tests of Within-Subjects Effects table.

Figura 41 – Teste de Esfericidade considerando o nível de significância de $p=0,5$ e o coeficiente de confiança de 95,5%. Observa-se que $\text{Sig.} > 5\%$. Fonte: O autor.

Como este teste resultou em um valor de significância maior que 5%, $p > 0,05$, aceitou-se a hipótese nula H0. E tanto no teste realizado com os dados obtidos via TM-902C, quanto no teste com os dados obtidos via ADS1118, aceita-se a hipótese H0, pois em ambos, o valor de significância é maior que 5%, conforme a Fig.(41). Isso quer dizer que, em cada método comparado, existe pelo menos uma diferença entre as médias das temperaturas medidas.

Isso influenciou na interpretação de outras tabelas de comparação, como a tabela de comparação par a par (*Pair Wise Comparisons*) gerada pelo IBM SPSS, as quais permitem verificar onde ocorrem as principais diferenças, o efeito do tempo e de interação entre tempo e tipo de medidor, sob as medidas.

5.2.3 ANOVA e o teste complementar - *Post-Hoc*

Foi verificado que existe distanciamento entre as médias dos grupos, variação dentro de cada grupo, e que existem valores que pertencem à diferentes conjuntos de temperatura que ocupam a mesma faixa.

Como existem diferentes faixas de temperatura, os valores medidos podem estar próximos, afastados ou entre as temperaturas conhecidas. Para distinguir os grupos aos quais determinados valores de temperatura pertencem, o teste de *Post-Hoc Sidak* foi realizado, com base na Fig.(42), a qual apresenta o resultado do teste *Pairwise Comparisons* gerado no SPSS.

Pairwise Comparisons						
Measure: MEASURE_1						
(I) Tempo		Mean Difference (I-J)	Std. Error	Sig. ^b	95% Confidence Interval for Difference ^b	
					Lower Bound	Upper Bound
1	2	1,721 [*]	,619	,027	,161	3,281
	3	1,309	,918	,415	-1,005	3,623
2	1	-1,721 [*]	,619	,027	-3,281	-,161
	3	-,412	,826	,046	-2,492	1,668
3	1	-1,309	,918	,415	-3,623	1,005
	2	,412	,826	,046	-1,668	2,492
Based on estimated marginal means						
*. The mean difference is significant at the ,05 level.						
b. Adjustment for multiple comparisons: Sidak.						

Figura 42 – Teste *Pairwise Comparisons*. IBM SPSS. Nível de significância de $p=0,5$ e coeficiente de confiança de 95,5%. Fonte: O autor.

Para avaliar a significância das diferenças observadas entre os dados, foram consideradas as seguintes hipóteses:

H0: $p > 0,05$ | Não existe diferença estatisticamente significativa entre os valores de temperatura

H1: $p < 0,05$ | Existe diferença estatisticamente significante entre os valores de temperatura

A ANOVA com medidas repetidas mostrou que existe o efeito do tempo e de cada tipo de medidor de temperatura sob os conjuntos de valores medidos. Os gráficos mostraram, de forma simplificada, as diferenças entre os conjuntos de valores de temperatura medidos e o quanto se distanciam dos valores esperados.

Desta forma, na Fig.(43) referente as medidas do Grupo 1, observa-se que as diferenças entre as medidas de temperatura na faixa de 25 °C e 415 °C, medidas por meio do medidor TM-902C, não são estatisticamente significativas, o mesmo foi verificado na faixa de temperatura de 475 °C e 490 °C.

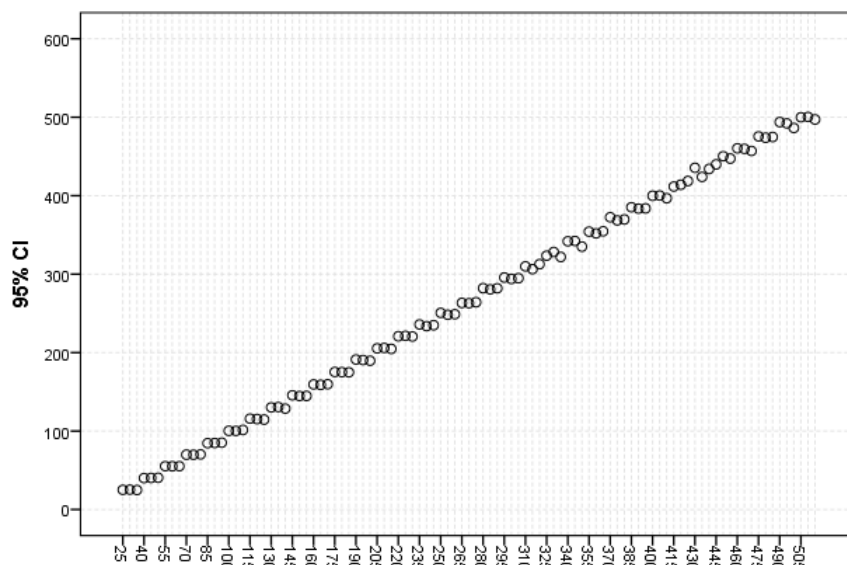


Figura 43 – Gráfico de medidas do Grupo 1 entre 25 °C e 505 °C - Post Hoc - SPSS. Fonte: O autor.

Enquanto na Fig. (44), relacionada às medidas do Grupo 2 para a mesma faixa de valores, as medidas realizadas por meio do sensor ADS1118, mostraram-se mais dispersas e distantes dos valores de temperatura de referência.

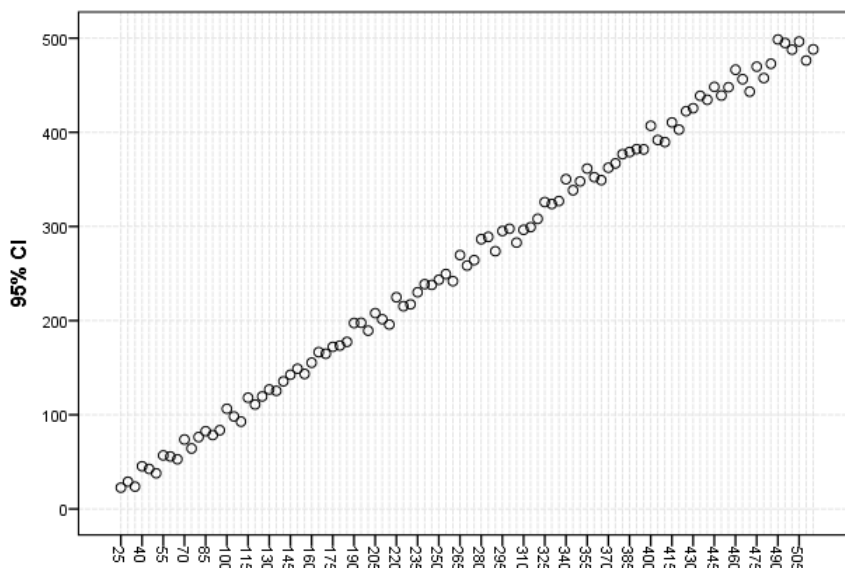


Figura 44 – Gráfico de medidas do Grupo 2 entre 25 °C e 505 °C - Post Hoc - SPSS. Fonte: O autor.

Nos casos em que p foi menor que 0,05, existe efeito do tempo e da interação entre tempo e tipo de medidor. Analisando o teste *Post-Hoc Sidak* da interação entre tempo e tipo de medidor, verifica-se que há efeito do tempo [$F(1,696, 63,048) = 2,539$; $p < 0,05$], que existiu diferença entre as medidas realizadas nos intervalos de tempo t_1 e t_2 e a diferença aumentou entre t_2 e t_3 . Isso pode ser visualizado no gráfico da Fig.(45), a qual

mostra que o desempenho do TM-902C se manteve superior durante o ensaio de medidas de temperatura.

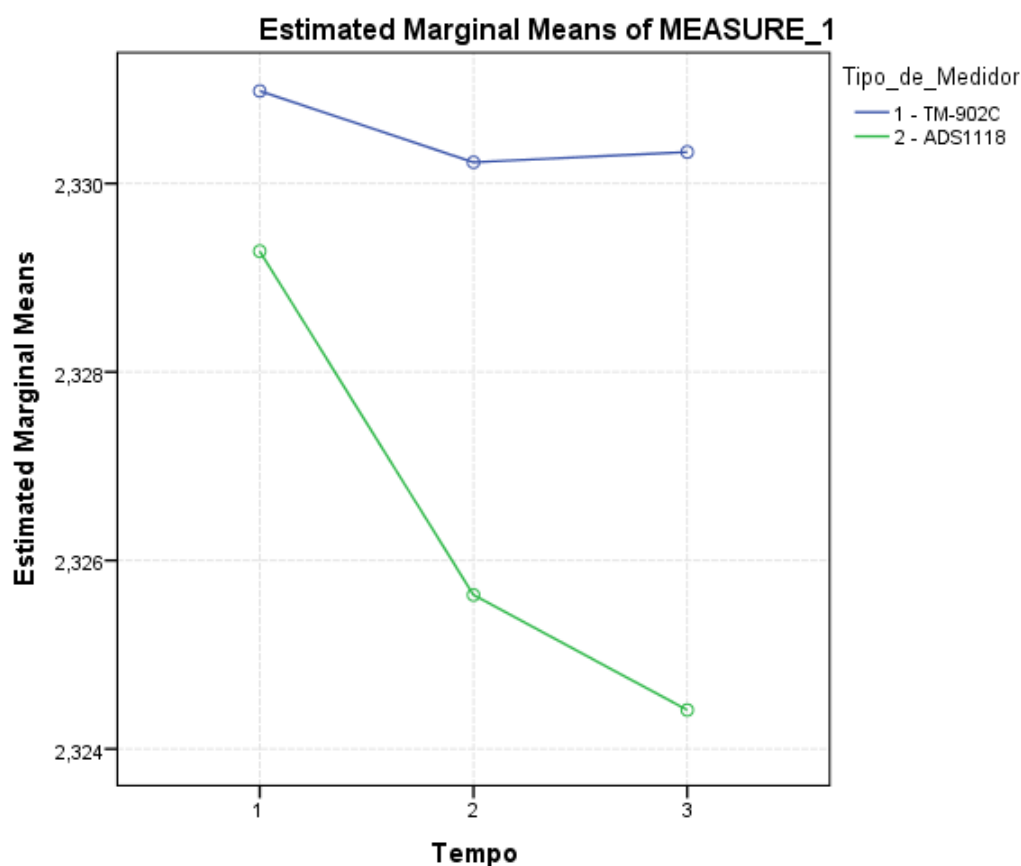


Figura 45 – Gráfico de comparação das médias dos Grupos 1 e 2 - *Post Hoc Sidak* - SPSS.
Fonte: O autor.

5.3 *Bland-Altman*

Observou-se que existe uma correlação entre os métodos de medição utilizados nos dispositivos comparados, tendo em vista que os valores medidos são diretamente proporcionais. Entretanto, como não ficou claro se os métodos concordam entre si, optou-se por realizar o teste de *Bland-Altman*.

O teste de *Bland-Altman* permite avaliar a concordância entre as metodologias utilizadas pelos dispositivos comparados, aos quais medem o mesmo parâmetro, a temperatura.

Tomando as medidas de temperatura, de cada tipo de medidor, registradas no banco de dados, foram extraídas as diferenças e médias para cada valor de temperatura de referência, conforme a Fig. (46).

Referência [°C]	Medida t1 - AD51118_RPi [°C]	Medida t1 TM-902C [°C]	Diferenças	Média		
25	22,6	25,1	-2,5	23,9		
40	45,3	40	5,3	42,7		
55	56,8	55,3	1,5	56,1		
70	73,7	69,9	3,8	71,8		
85	82,5	84,5	-2	83,5		
100	106,4	100,3	6,1	103,4		
115	118,1	115,8	2,3	117,0		
130	126,9	130,2	-3,3	128,6		
145	142,5	145,6	-3,1	144,1		
160	155,3	159,4	-4,1	157,4		
175	172,2	175,3	-3,1	173,8		
190	197,4	191,2	6,2	194,3		
205	208	205,4	2,6	206,7		
220	224,8	220,8	4	222,8		
235	230,1	235,9	-5,8	233,0		
250	243,3	250,7	-7,4	247,0		
265	269,6	263,3	6,3	266,5		
280	286,5	282,1	4,4	284,3		
295	295,1	295,7	-0,6	295,4		
310	296,4	309,9	-13,5	303,2		
325	325,9	323,6	2,3	324,8		
340	350,2	341,9	8,3	346,1		
355	361,5	354,2	7,3	357,9		
370	362,4	372,5	-10,1	367,5		
385	379,1	385,3	-6,2	382,2		
400	406,9	399,9	7	403,4		
415	410,5	411,7	-1,2	411,1		
430	425,6	435,6	-10	430,6		
445	448,3	439,9	8,4	444,1		
460	466,5	460,4	6,1	463,5		
475	469,8	475,6	-5,8	472,7		
490	498,6	493,8	4,8	496,2		
505	496,4	499,9	-3,5	498,2		
			Média das diferenças	0,14	SUP	11,78
			DP	5,94	INF	-11,51

Figura 46 – Planilha de análise de dados - Medidas de temperatura obtidas durante o ensaio com os dois tipos de medidores - Diferenças e médias obtidas para encontrar a média das diferenças, o desvio padrão e os limites inferior e superior do gráfico de *Bland-Altman*. Fonte: O autor.

Para fins de análise, estes dados foram exportados para o ambiente de análise IBM SPSS e em seguida, foi realizado o teste T para cada grupo, com o intuito de verificar se as diferenças entre as medidas são estatisticamente diferentes de zero. A Figura (47) mostra uma tabela com este resultado.

One-Sample Test						
	Test Value = 0					
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
Diferenças	,089	32	,930	,091	-2,00	2,18

Figura 47 – Tabela com o resultado do teste T . Nível de significância de $p=0,5$ e coeficiente de confiança de 95,5%. Fonte: O autor.

Utilizando um intervalo de confiança de 95%, como o valor de de significância foi

superior a 5%, $p > 0,05$, a diferença não foi significativamente diferente de zero, portanto, concluiu-se que existe concordância entre os dois métodos.

A Figura (48), revela a média das diferenças, a qual corresponde a linha central do gráfico, sendo o padrão esperado de diferença entres os resultados.

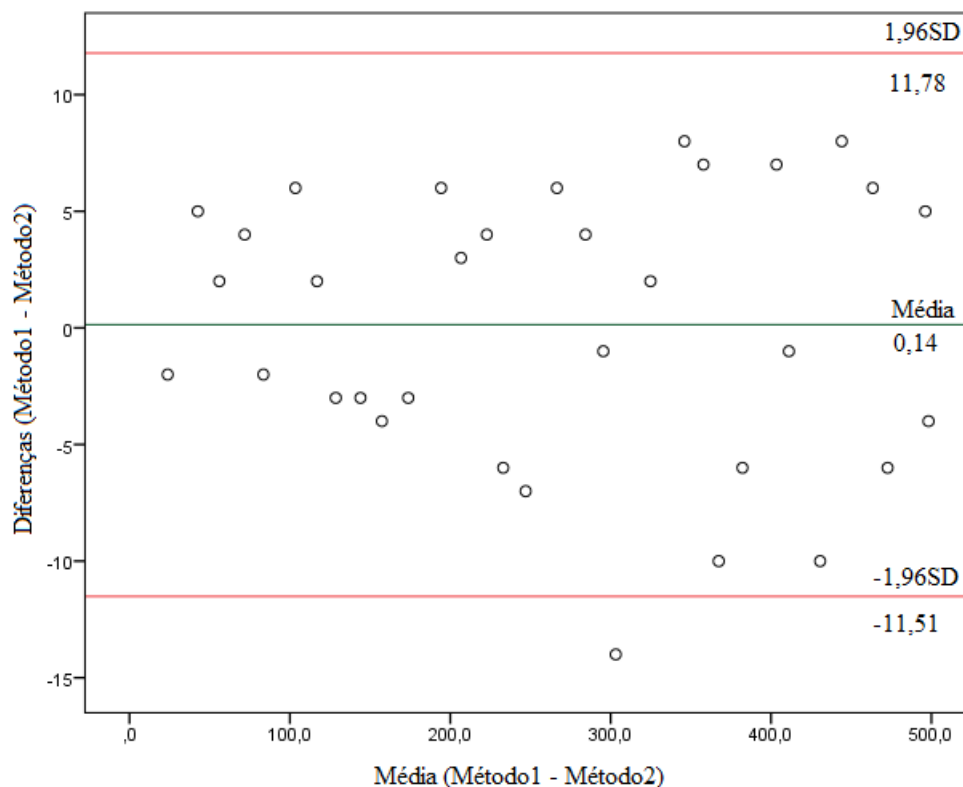


Figura 48 – Gráfico de diferença das médias - *Bland-Altman* - SPSS. Fonte: O autor.

O desvio padrão das diferenças está diretamente relacionado ao intervalo de confiança de 95%, a qual está a $1,96DP$ acima da média e $-1,96DP$ abaixo da média.

Tem-se o valor da média da diferença entre os dois métodos (linha verde), que como está próxima de zero, denota-se que os métodos são concordantes entre si.

Verifica-se que alguns valores se distanciam da linha de média das diferenças, entretanto, na maioria dos casos estão dentro do intervalo de confiança de 95%.

Para verificar se existe um viés, nessa análise, foi necessário saber se ocorre uma tendencia a ter valores de diferença acima ou abaixo da média das diferenças. Portanto, foi realizado um teste de regressão linear considerando-se as diferenças como variáveis dependentes e a média entre as medidas como variáveis independentes. O resultado deste teste pode ser observado na Fig.(49).

Coefficients ^a					
Model		Unstandardized Coefficients		Standardized Coefficients	Sig.
		B	Std. Error	Beta	
1	(Constant)	,771	2,197		,351
	Média	-,003	,007	-,063	,728

a. Dependent Variable: Diferenças

Figura 49 – Tabela com o resultado do teste de regressão linear, para verificar se existe um viés nos dados analisados - SPSS. Nível de significância de $p=0,5$ e coeficiente de confiança de 95,5%. Fonte: O autor.

Estabelecendo-se as seguintes hipóteses:

H0: $p < 0,05$ | Existe viés de proporção

H1: $p > 0,05$ | Não existe viés de proporção

Como o valor de significância p obtido para a média foi superior a 5%, constatou-se que não existe um viés de proporção. Isso significa que os valores se distribuem de maneira homogênea acima e abaixo da média das diferenças entre os dois métodos. Logo, os valores das diferenças não apresentaram tendência a se concentrar apenas acima ou abaixo do valor médio da diferença entre os dois métodos.

6 CONSIDERAÇÕES FINAIS

Neste projeto, foi realizada a implementação de um dispositivo medidor de temperatura, empregando-se uma plataforma *Raspberry Pi* e um módulo *TI 430BOOST-ADS1118* com termopar do tipo K. O dispositivo apresentou resultados relativamente aproximados aos valores medidos por meio do dispositivo comercial TM-902C, permitindo o acompanhamento e registro das medidas de temperatura e horários, via *Wi-Fi*, por meio de *smartphone* ou computador pessoal.

A fundamentação teórica para a conversão de temperatura utilizada no dispositivo implementado parte dos mesmos princípios utilizados em dispositivos comerciais, contudo os métodos de implementação são distintos, bem como as respectivas formas de leitura e registro. Para validar o sistema desenvolvido e comparar sua performance com aquela apresentada pelos métodos tradicionais, foi realizado um ensaio de medidas de temperatura e, de posse dos dados obtidos, foram realizados vários testes, como o teste ANOVA que permitiu verificar que existem diferenças entre e intra grupos de medidas de temperatura. E o teste de *Bland-Altman* ao qual mostrou que os métodos concordam entre si e as medidas obtidas não foram enviesadas.

Dada a importância do emprego de termopares em diversos experimentos laboratoriais e em processos de controle de qualidade, verificou-se que recursos de registro automático (*data logger*) e remoto de medidas podem proporcionar maior comodidade e economia de tempo ao operador. O dispositivo atendeu aos respectivos requisitos durante a realização dos ensaios, no entanto, foram identificadas algumas limitações, como erro sistemático nas medidas e atraso no tempo de escrita de dados no arquivo .csv.

A estimativa do custo de implementação deste projeto, quando comparado ao preço de produtos comerciais que dispõem de ferramentas de registro e monitoramento de temperatura via *Wi-Fi*, mostrou que é possível economizar recurso financeiro. No entanto, deve-se levar em conta que por se tratar de um projeto em fase inicial, existe a necessidade realizar melhorias principalmente na implementação do algoritmo de conversão de valores de tensão em temperatura, da camada de comunicação em tempo real e do ajuste de *offset* em faixas menores de medição, para que seja possível uma justa comparação dessa relação de custo-benefício.

Destarte, é importante frisar que, embora o registro automático de dados evite erros que causam imprecisão nas medidas, neste projeto, faz-se necessário realizar análises mais criteriosas, calibração e ajuste de *offset* via *software*, de modo que as medições tornem-se mais próximas dos valores de referência.

6.1 Proposta de melhorias

Pode-se sugerir que em trabalhos futuros realize-se:

- Medidas em faixas menores de temperatura, com a finalidade de descobrir o impacto real de eventuais comportamentos não lineares e parâmetros como o erro de deslocamento de *bits* e erro de ganho nos valores medidos pelo ADS1118;
- Estudos mais aprofundados em relação à calibração, utilizando-se ferramentas de análise no domínio da frequência;
- Análise de *offset* em diferentes intervalos de temperatura, por meio da comparação das medidas utilizando-se um dispositivo de referência previamente calibrado, com a finalidade de fornecer os respectivos valores de ajuste para compensação de erro sistemático. E a implementação de um módulo que automatize o processo de ajuste de *offset* via *software*;
- Na implementação *web*, o desenvolvimento de novos blocos com funções que permitam ao usuário selecionar a faixa de medidas desejada, inserir o valor de ajuste de *offset* e plotar os dados de medidas de temperatura, em tempo real, para facilitar a interpretação de experimentos;
- Projeto de um recipiente que possua blindagem específica com a finalidade de cobrir o sistema e reduzir efeitos de EMI.

Referências

- AGUIRRE, L. A. Controle de sistemas amostrados. In: UFMG (Ed.). Belo Horizonte, Brasil: [s.n.], 2019. p. 25–78. Citado na página 28.
- BECKMAN, M.; CHIOYE, L. *Precision Thermocouple Measurement with the ADS1118*. [S.l.], 2011. 11 p. Disponível em: <<https://www.ti.com/lit/ds/sbas457f/sbas457f.pdf>>. Citado 5 vezes nas páginas 7, 10, 37, 38 e 79.
- BURNS, G. et al. Temperature-electromotive force reference functions and tables for the letter-designated thermocouple types based on the ITS-90. In: NIST (Ed.). Gaithersburg, USA: [s.n.], 1993. p. 153–187. Citado 2 vezes nas páginas 12 e 25.
- BURNS, G. W.; SCROGER, M. G. Nist measurement services: The calibration of thermocouples and thermocouple materials. In: NIST (Ed.). Washington, USA: [s.n.], 1989. p. 2–189. Citado 3 vezes nas páginas 12, 25 e 26.
- CHANDLER, D. L. *MIT News Office - Explained: Thermoelectricity*. [S.l.], 2010. Disponível em: <<http://news.mit.edu/2010/explained-thermoelectricity-0427>>. Citado na página 24.
- FIALHO, A. B. Instrumentação industrial conceitos, aplicações e análises. In: ERICA (Ed.). [S.l.: s.n.], 2002. p. 100–150. Citado na página 18.
- FILIPEFLOP. *FilipeFlop Componentes Eletrônicos*. [S.l.], 2020. Disponível em: <<https://www.filipeflop.com/>>. Citado 4 vezes nas páginas 7, 12, 28 e 30.
- FOUNDATION, R. P. *Raspberry Pi Zero W*. [S.l.], 2017. Disponível em: <<https://www.raspberrypi.org/products/raspberry-pi-zero-w/>>. Citado 2 vezes nas páginas 7 e 35.
- GIL, A. C. Como elaborar projetos de pesquisa (4ª ed.). In: ATLAS (Ed.). Brasil, SP: [s.n.], 2002. p. 23–166. Citado 2 vezes nas páginas 21 e 30.
- HIRAKATA, V. N.; CAMEY, S. A. Análise de concordância entre métodos de bland-altman. In: *Revista HCPA*. [S.l.: s.n.], 2009. v. 29, p. 261–268. Citado na página 31.
- IBMKNOWLEDGECENTER. *Mauchly's Test of Sphericity*. [S.l.], 2019. Disponível em: <https://www.ibm.com/support/knowledgecenter/SSLVMB_27.0.0/statistics_casestudies_project_ddita/spss/tutorials/glmlr_testmarket_mauchly.html>. Citado na página 31.
- INMETRO. *Later (Laboratório de Termometria)*. [S.l.], 2020. Disponível em: <<http://www.inmetro.gov.br/laboratorios/servicos/calibDquim.asp#LATER>>. Citado na página 49.
- INSTRUMENTS, T. *430BOOST ADS1118 Booster Pack - Display LCD: User's guide*. [S.l.], 2013. Disponível em: <<http://www.ti.com/lit/ug/sbau207/sbau207.pdf>>. Citado 2 vezes nas páginas 7 e 39.

- INSTRUMENTS, T. *430BOOST ADS1118 - Evaluation Module - Temperature Measurement*. [S.l.], 2019. Disponível em: <<http://www.ti.com/tool/430BOOST-ADS1118?keyMatch=430BOOST-ADS1118%20&tisearch=Search-EN-everything&usecase=part-number>>. Citado 4 vezes nas páginas 7, 8, 36 e 42.
- INSTRUMENTS, T. *The Engineer's Guide to Temperature Sensing*. [S.l.], 2019. Disponível em: <<https://www.ti.com/sensors/temperature-sensors/overview.html>>. Citado na página 18.
- JUCA, S.; PEREIRA, R. Aplicações práticas de sistemas embarcados: Linux utilizando raspberry pi. In: POD (Ed.). Rio de Janeiro, Brasil: [s.n.], 2018. v. 1, p. 216. Disponível em: <<https://podeditora.com.br/wp-content/uploads/2019/03/Livro-Applicacoes-Praticas-Rpi-Pod-site.pdf>>. Citado na página 30.
- LERCH, B.; NATHAL, M.; KELLER, D. Thermocouple calibration and accuracy in a materials testing laboratory. In: NASA/TM-2002-211507 (Ed.). Cleveland, Ohio: [s.n.], 2002. p. –. Citado na página 61.
- MAKRIS, J. P. et al. Studying and calibrating thermocouples: A laboratory exercise at the technological educational institute of crete. In: LBUS (Ed.). Sibiu, Romênia: [s.n.], 2003. p. 2–5. Citado na página 18.
- MATHWORKS. *isoutlier - Find outliers in data*. [S.l.], 2020. Disponível em: <<https://www.mathworks.com/help/matlab/ref/isoutlier.html>>. Citado 3 vezes nas páginas 11, 31 e 85.
- MIOT, H. A. Avaliação da normalidade dos dados em estudos clínicos e experimentais. In: *J Vasc Bras - Scielo*. [S.l.: s.n.], 2017. p. 88–91. Citado na página 31.
- MORAES, A. F. Redes sem fio: Instalação, configuração e segurança - fundamentos. In: ÉRICA (Ed.). São Paulo, Brazil: [s.n.], 2010. p. 63–75. Citado na página 29.
- MORRIS, A. S. Measurement and instrumentations principles (3^a ed.). In: BUTTERWORTH-HEINEMANN (Ed.). Oxford, UK: [s.n.], 2001. p. 296–310. Citado na página 18.
- NIST. *National Institute of Standards and Technology (NIST) - Tables of thermoelectric voltages of each type by temperature range, of the coefficients, or of the inverse coefficients*. [S.l.], 2014. Disponível em: <<https://srdata.nist.gov/its90/menu/menu.html>>. Citado 3 vezes nas páginas 20, 47 e 51.
- OLIVEIRA, C. et al. Sensor de temperatura via raspberry para transporte de refrigerados. In: TECNOLOGICAS, I. C. E. e (Ed.). Aracaju, Brazil: [s.n.], 2018. p. 19–26. ISSN 2359-4942. Citado na página 29.
- OMEGA. *Monitor de Temperatura com Comunicação via Internet. OMEGA a Spectris Company*. [S.l.], 2020. Disponível em: <<https://br.omega.com/pptst/ISD-TC.html>>. Citado na página 19.
- PI4J. *The pi4j project - Pin Numbering: Raspberry pi zero w*. [S.l.], 2012. Disponível em: <<https://pi4j.com/1.2/pins/model-zero-w-rev1.html>>. Citado 2 vezes nas páginas 7 e 35.

POLLOCK, D. D. Thermocouples theory and properties. In: PRESS, C. (Ed.). New York, USA: [s.n.], 1991. p. 138–155. Citado na página 24.

RIVERA, A. D. Redes de equipamentos sem fio de uso pessoal: Comparação de tecnologias emergentes e análise de tendências: Tese (mestrado em engenharia de sistemas eletrônicos). In: USP (Ed.). São Paulo, Brazil: [s.n.], 2010. p. 53–77. Citado na página 29.

THOMAZINI, D.; ALBUQUERQUE, P. U. B. Sensores industriais fundamentos e aplicações (8ª ed.). In: ÉRICA (Ed.). [S.l.: s.n.], 2011. p. 93–100. Citado 2 vezes nas páginas 18 e 24.

VOLBRECHT, A. *Guidelines For Good Thermocouple Pratices*. [S.l.], 2007. Disponível em: <<https://www.asminternational.org/documents/10192/1915068/http00703p048.pdf/236107c7-ec38-40e6-8ecc-51a23dfefaa4>>. Citado 2 vezes nas páginas 12 e 25.

WEBSTER, J. G. The measuriment instrumentation and sensors handbook. In: PRESS, C. (Ed.). New York, USA: [s.n.], 2000. p. 992–1010. Citado na página 28.

WILSON, J. S. Sensor technology handbook. In: ELSEVIER (Ed.). Oxford, UK: [s.n.], 2005. p. 531–599. Citado na página 18.

YOUSAF, S. *Temperature Measurement For Lab And Science Projects, Element14 Tutorials*. [S.l.], 2014. Disponível em: <<https://www.element14.com>>. Citado 2 vezes nas páginas 51 e 80.

ZUMBAHLEN, H. Linear circuit design handbook. In: ELSEVIER (Ed.). Burlington, USA: [s.n.], 2008. p. 215–223. Citado 3 vezes nas páginas 7, 26 e 27.

Anexos

ANEXO A – Primeiro Anexo

Mapa de registro e configuração do ADS1118, disponibilizado pela *Texas Instruments*, usado como base para recuperar as medições de temperatura por meio da RPi-zero-Wh.

Bit	Field	Type	Reset	Description
15	SS	R/W	0h	Single-shot conversion start This bit is used to start a single conversion. SS can only be written when in power-down state and has no effect when a conversion is ongoing. When writing: 0 = No effect 1 = Start a single conversion (when in power-down state) Always reads back 0 (default).
14:12	MUX[2:0]	R/W	0h	Input multiplexer configuration These bits configure the input multiplexer. 000 = AIN _P is AIN0 and AIN _N is AIN1 (default) 001 = AIN _P is AIN0 and AIN _N is AIN3 010 = AIN _P is AIN1 and AIN _N is AIN3 011 = AIN _P is AIN2 and AIN _N is AIN3 100 = AIN _P is AIN0 and AIN _N is GND 101 = AIN _P is AIN1 and AIN _N is GND 110 = AIN _P is AIN2 and AIN _N is GND 111 = AIN _P is AIN3 and AIN _N is GND
11:9	PGA[2:0]	R/W	2h	Programmable gain amplifier configuration These bits configure the programmable gain amplifier. 000 = FSR is $\pm 6.144 \text{ V}^{(1)}$ 001 = FSR is $\pm 4.096 \text{ V}^{(1)}$ 010 = FSR is $\pm 2.048 \text{ V}$ (default) 011 = FSR is $\pm 1.024 \text{ V}$ 100 = FSR is $\pm 0.512 \text{ V}$ 101 = FSR is $\pm 0.256 \text{ V}$ 110 = FSR is $\pm 0.256 \text{ V}$ 111 = FSR is $\pm 0.256 \text{ V}$
8	MODE	R/W	1h	Device operating mode This bit controls the ADS1118 operating mode. 0 = Continuous conversion mode 1 = Power-down and single-shot mode (default)
7:5	DR[2:0]	R/W	4h	Data rate These bits control the data rate setting. 000 = 8 SPS 001 = 16 SPS 010 = 32 SPS 011 = 64 SPS 100 = 128 SPS (default) 101 = 250 SPS 110 = 475 SPS 111 = 860 SPS
4	TS_MODE	R/W	0h	Temperature sensor mode This bit configures the ADC to convert temperature or input signals. 0 = ADC mode (default) 1 = Temperature sensor mode
3	PULL_UP_EN	R/W	1h	Pullup enable This bit enables a weak internal pullup resistor on the DOUT/DRDY pin only when CS is high. When enabled, an internal 400-k Ω resistor connects the bus line to supply. When disabled, the DOUT/DRDY pin floats. 0 = Pullup resistor disabled on DOUT/DRDY pin 1 = Pullup resistor enabled on DOUT/DRDY pin (default)
2:1	NOP[1:0]	R/W	1h	No operation The NOP[1:0] bits control whether data are written to the Config register or not. For data to be written to the Config register, the NOP[1:0] bits must be '01'. Any other value results in a NOP command. DIN can be held high or low during SCLK pulses without data being written to the Config register. 00 = Invalid data, do not update the contents of the Config register 01 = Valid data, update the Config register (default) 10 = Invalid data, do not update the contents of the Config register 11 = Invalid data, do not update the contents of the Config register
0	Reserved	R	1h	Reserved Writing either 0 or 1 to this bit has no effect. Always reads back 1.

(1) This parameter expresses the full-scale range of the ADC scaling. No more than VDD + 0.3 V must be applied to this device.

Figura 50 – Tabela de configuração do ADS1118. Fonte: (BECKMAN; CHIOYE, 2011).

ANEXO B – Segundo Anexo

O desenvolvimento da aplicação *web* foi iniciada e estão disponibilizados *links* específicos de alguns scripts empregados na implementação da página utilizando HTML, CCS e *JavaScript*, com base no pseudocódigo disponibilizado pela *Texas Instruments* e no projeto criado por Yousaf (YOUSAF, 2014), a qual foi adaptado para recuperar as medições de temperatura usando a RPi zero Wh.

Tabela 6 – Links disponíveis

Descrição	Link
Códigos adaptados	< https://github.com/filipe19/Dispositivo_de_Monitoramento_de_Temperatura_Via_Wi-Fi >
Código Fonte	< https://www.ti.com/product/ADS1118 >
Projeto de referência	< https://github.com/filipe19/TemperatureMeasurement >

```

1 // Trechos do código adaptado --- Sensor de temperatura ADS1118
2
3 // Funções
4 int delay_ms(unsigned int msec)
5 {
6     int ret;
7     struct timespec a;
8     if (msec>999)
9     {
10         msec=999;
11     }
12     a.tv_nsec=((long)(msec))*1E6d;
13     a.tv_sec=0;
14     return(0);
15 }
16
17 void setup_io()// Configura uma região de memória para acesso GPIO
18 {
19     /* open /dev/mem */
20     if ((mem_fd = open("/dev/mem", O_RDWR|O_SYNC) ) < 0) {
21         printf("can't open /dev/mem \n");
22         exit(-1);
23     }
24
25     /* mmap GPIO */
26     gpio_map = mmap(
27         NULL,                // Qualquer endereço
28         BLOCK_SIZE,          // Tamanho mapeado
29         PROT_READ|PROT_WRITE, // Habilita leitura e escrita para a memória mapeada
30         MAP_SHARED,           // Compartilha com outros processos
31         mem_fd,               // Arquivo para mapear
32         GPIO_BASE
33     );
34
35     close(mem_fd); //Não é necessário manter mem_fd aberto após mmap
36
37     if (gpio_map == MAP_FAILED) {
38         printf("mmap error %d\n", (int)gpio_map);
39         exit(-1);
40     }
41
42     gpio = (volatile unsigned *)gpio_map; // Uso de ponto volátil
43 }
44
45 int spi_open(int* f_desc, int sel, uint8_t config) // Habilita o barramento SPI
46 {
47     uint8_t spi_bits = 8;
48     int ret;
49
50     if (sel)
51         *f_desc=open(device1, O_RDWR);
52     else
53         *f_desc=open(device0, O_RDWR);
54     if (*f_desc<0)
55     {
56         fprintf(stderr, "Error opening device: %s\n", strerror(errno));
57         return(-1);
58     }
59     ...
60     return(0);
61 }
62
63 void lcd_writecom(unsigned char c)// Comandos para o LCD
64 {
65     int ret;
66
67     GPIO_CLR = 1<<LCD_RS_GPIO; // Define RS baixo para o comando transmitido
68
69     txbuf[0]=c;
70     spi.len=1;
71     spi.delay_usecs=0;
72     spi.speed_hz=spi_speed;
73     spi.bits_per_word=spi_bits;
74     spi.cs_change=0;
75     spi.tx_buf=(unsigned long)txbuf;
76     spi.rx_buf=(unsigned long)rxbuf;
77
78     ret=ioctl(lcd_fd, SPI_IOC_MESSAGE(1), &spi);
79     if (ret<0)
80     {
81         fprintf(stderr, "Error performing SPI exchange: %s\n", strerror(errno));
82         exit(1);
83     }
84 }
85
86 void lcd_writedata(unsigned char c) // Escrita dos dados
87 {
88     int ret;
89
90     GPIO_SET = 1<<LCD_RS_GPIO; // Define RS alto para escrita de dados
91
92     txbuf[0]=c;
93 }

```

Figura 51 – Código em C, adaptado. Implementado para recuperar valores de conversão de FEM em tensão por meio da interface entre a *Raspberry Pi Zero W* e o ADS1118. Fonte: Yousef, adaptado.

```

97 spi.bits_per_word=spi_bits;
98 spi.cs_change=0;
99 spi.tx_buf=(unsigned long)txbuf;
100 spi.rx_buf=(unsigned long)rxbuf;
101
102 ret=ioctl(lcd_fd, SPI_IOC_MESSAGE(1), &spi);
103 if (ret<0)
104 {
105     fprintf(stderr, "Error performing SPI exchange: %s\n", strerror(errno));
106     exit(1);
107 }
108
109 void lcd_clear(void) // Limpar LCD
110 {
111     lcd_writecom(0x01);
112     delay_ms(2);
113     lcd_writecom(0x02);
114     delay_ms(2);
115 }
116
117 void lcd_display_string(unsigned char line_num, char *ptr) //Escreve uma string na linha do LCD
118 {
119     if(line_num==0) // Primeira linha
120     {
121         lcd_writecom(0x80);
122     }
123     else if(line_num==1) // Segunda linha
124     {
125         lcd_writecom(0xc0);
126     }
127     while (*ptr)
128     {
129         lcd_writedata(*ptr++);
130     }
131 }
132
133 // inicializa e limpa o display
134 void lcd_init(void)
135 {
136     GPIO_SET = 1<<LCD_RS_GPIO;
137     lcd_writecom(0x30); // inicia
138     lcd_writecom(0x39); // define função
139     lcd_writecom(0x14); // oscilador de frequencia interno
140     lcd_writecom(0x56); // controle liga desliga
141     lcd_writecom(0x6D); // controle seguidor
142     lcd_writecom(0x70); // contraste
143     lcd_writecom(0x0C); // display ligado
144     lcd_writecom(0x06); // modo de entrada
145     lcd_writecom(0x01); // limpa
146     delay_ms(20);
147 }
148
149 // Envia quatro bytes (dois bytes de configuração repetidos uma segunda vez, e retorna dois bytes
150 int therm_transact(void)
151 {
152     int ret;
153     txbuf[0]=txbuf[0] | 0x80;
154     spi.len=4;
155     txbuf[2]=txbuf[0];
156     txbuf[3]=txbuf[1];
157     spi.delay_usecs=0;
158     spi.speed_hz=spi_speed;
159     spi.bits_per_word=spi_bits;
160     spi.cs_change=0;
161     spi.tx_buf=(unsigned long)txbuf;
162     spi.rx_buf=(unsigned long)rxbuf;
163     if (DBG_PRINT)
164     printf("sending [%02x %02x %02x %02x]. ", txbuf[0], txbuf[1], txbuf[2], txbuf[3]);
165     ret=ioctl(ads_fd, SPI_IOC_MESSAGE(1), &spi);
166     if (ret<0)
167     {
168         fprintf(stderr, "Error performing SPI exchange: %s\n", strerror(errno));
169         exit(1);
170     }
171     if (DBG_PRINT)
172     printf("received [%02x %02x]\n", rxbuf[0], rxbuf[1]);
173     ret=rxbuf[0];
174     ret=ret<<8;
175     ret=ret | rxbuf[1];
176     return(ret);
177 }
178
179 // Transforma o código do sensor de temperatura interna para o de compensação
180 //ao qual ainda é adicionado o código do termopar
181 int local_compensation(int local_code)
182 {
183     float tmp,local_temp;
184     int comp;
185     local_code = local_code / 4;
186     local_temp = (float)local_code / 32;
187     ...
188     else if (local_temp> 50 && local_temp <=60) //50-60
189     {
190         tmp = (0x0035*(local_temp - 50))/10 + 0x0103;
191         comp = tmp;
192     }
193     else if (local_temp> 60 && local_temp <=80) //60-80
194     {
195         tmp = (0x006A*(local_temp - 60))/20 + 0x0138;
196         comp = tmp;
197     }
198     else if (local_temp> 80 && local_temp <=125) //80-125
199     {
200         tmp = (0x00EE*(local_temp - 80))/45 + 0x01A2;
201         comp = tmp;
202     }
203     ...
204 }

```

Figura 52 – Código em C, adaptado. Implementado para recuperar valores de conversão de FEM em tensão por meio da interface entre a *Raspberry Pi Zero W* e o ADS1118. Fonte: Yousef, adaptado.

```

213     else
214     {
215         comp = 0;
216     }
217     return comp;
218 }
219
220 // Convertendo resultados do conversor AD para temperatura
221 int adc_code2temp(int code)
222 {
223     float temp;
224     int t;
225
226     temp = (float)code;
227
228
229     if (code > 0xFCC6 && code < 0xFCC8) temp = (float)(10*(temp-0xFCC6)) / 0x0002 -270.0f; // -270 C até -260 C
230     ...
231     else if (code > 0x0066 && code < 0x0099) temp = (float)(10*(temp-0x0066)) / 0x0033 + 20.0f; // 20 C até 30 C
232     else if (code > 0x0099 && code < 0x00CE) temp = (float)(10*(temp-0x0099)) / 0x0034 + 30.0f; // 30 C até 40 C
233     else if (code > 0x00CE && code < 0x0102) temp = (float)(10*(temp-0x00CE)) / 0x0034 + 40.0f; // 40 C até 50 C
234     else if (code > 0x0102 && code < 0x0137) temp = (float)(10*(temp-0x0102)) / 0x0034 + 50.0f; // 50 C até 60 C
235     ...
236     else if (code > 0x09AE && code < 0x09E5) temp = (float)(10*(temp-0x09AE)) / 0x0036 + 470.0f; // 470 C até 480 C
237     else if (code > 0x09E5 && code < 0x0A1B) temp = (float)(10*(temp-0x09E5)) / 0x0036 + 480.0f; // 480 C até 490 C
238     else if (code > 0x0A1B && code < 0x0A52) temp = (float)(10*(temp-0x0A1B)) / 0x0036 + 490.0f; // 490 C até 500 C
239     ...
240     else
241         return (int)(10*0x270F);
242
243     t = (int)(10*temp);
244
245     return t;
246 }
247
248 // Configurando e iniciando a conversão
249 void ads_config(unsigned int mode, unsigned int chan)
250 {
251     unsigned int tmp;
252     int ret;
253
254     if(chan)
255     {
256         if (mode==EXTERNAL_SIGNAL)
257             tmp = ADSCON_CH1;
258         else
259             tmp = ADSCON_CH1 + ADS1118_TS;
260     }
261
262     else
263     {
264         if (mode==EXTERNAL_SIGNAL)
265             tmp = ADSCON_CH0;
266         else
267             tmp = ADSCON_CH0 + ADS1118_TS;
268     }
269
270     txbuf[0]=(unsigned char)((tmp>>8) & 0xff);
271     txbuf[1]=(unsigned char)(tmp & 0xff);
272     ret=therm_transact();
273 }
274
275 // Lendo o resultado da conversão AD e iniciando uma nova conversão
276 int ads_read(unsigned int mode, unsigned int chan)
277 {
278     unsigned int tmp;
279     int result;
280
281     if(chan)
282     {
283         if (mode==EXTERNAL_SIGNAL)
284             tmp = ADSCON_CH1;
285         else
286             tmp = ADSCON_CH1 + ADS1118_TS;
287     }
288
289     else
290     {
291         if (mode==EXTERNAL_SIGNAL)
292             tmp = ADSCON_CH0;
293         else
294             tmp = ADSCON_CH0 + ADS1118_TS;
295     }
296
297     txbuf[0]=(unsigned char)((tmp>>8) & 0xff);
298     txbuf[1]=(unsigned char)(tmp & 0xff);
299     result=therm_transact();
300
301     return(result);
302 }
303
304 // Retornando a temperatura medida
305 double get_measurement(void)
306 {
307     int result;
308     int local_data;

```

Figura 53 – Código em C, adaptado. Implementado para recuperar valores de conversão de FEM em tensão por meio da interface entre a *Raspberry Pi Zero W* e o ADS1118. Fonte: Yousef, adaptado.


```

309     double result_d;
310
311     ads_config(INTERNAL_SENSOR,0); //inicia a medida do sensor de temperatura interna
312     delay_ms(10);
313     local_data=ads_read(EXTERNAL_SIGNAL,0); // ler a medição do sensor de temperatura interna e inicia o sensor de temperatura externa
314     delay_ms(10);
315     result=ads_read(EXTERNAL_SIGNAL,0); //ler a medição do sensor de temperatura externa e reinicia o sensor externo
316     local_comp = local_compensation(local_data);
317     result = result + local_comp;
318     result=result & 0xffff;
319     result = adc_code2temp(result);
320     result_d=((double)result)/10;
321     return(result_d);
322 }
323
324
325 void unixtime2string(char* int_part, char* out_time)
326 {
327     unsigned int ntime;
328     struct tm *nts;
329     char buf1[100];
330     char buf2[50];
331
332     sscanf(int_part, "%u", &ntime);
333     nts=localtime((time_t*)&ntime);
334     strftime(buf1, 100, "%H:%M:%S", nts);
335
336     strcpy(out_time, buf1);
337 }
338
339 void sig_handler(int signo)
340 {
341     if (signo == SIGINT)
342     {
343         printf("received SIGINT\n");
344         if (dofile)
345         {
346             fclose(outfile);
347         }
348         close(ads_fd);
349         lcd_clear();
350         lcd_display_string(1,"Bye");
351         close(lcd_fd);
352         printf("Bye\n");
353         exit(0);
354     }
355
356     ...
357     if (strcmp(argv[1], "lcdinit")==0) //inicializando o display LCD
358     {
359         spi_config=0;
360         ret=spi_open(&lcd_fd, 0, spi_config);
361         if (ret!=0)
362         {
363             printf("Exiting\n");
364             exit(1);
365         }
366         lcd_init();
367         close(lcd_fd);
368         exit(0);
369     }
370     if (strcmp(argv[1], "withtime")==0)
371     {
372         showtime=1;
373     }
374     else
375     {
376         sscanf(argv[1], "%d", &period); // periodo entre medidas
377         repeat=1;
378     }
379     if (argc>3)
380     {
381         if (strcmp(argv[2], "mag")==0) //imprimindo dados no LCD
382         {
383             spi_config=0;
384             ret=spi_open(&lcd_fd, 0, spi_config);
385             if (ret!=0)
386             {
387                 printf("Exiting\n");
388                 exit(1);
389             }
390             lcd_init();
391             lcd_display_string(0,argv[3]);
392             lcd_initialised=1;
393         }
394     }
395     if (argc>2)
396     {
397         strcpy(fname, argv[2]);
398         dofile=1;
399         if (argc>4)
400         {
401             if (strcmp(argv[3], "mag")==0)
402             {
403                 spi_config=0;
404                 ret=spi_open(&lcd_fd, 0, spi_config);
405                 if (ret!=0)
406                 {
407                     printf("Exiting\n");
408                     exit(1);
409                 }
410                 lcd_init();
411                 lcd_display_string(0,argv[4]);
412                 lcd_initialised=1;
413             }
414         }
415     }
416     ...
417     //Configuração SPI para o LCD
418     spi_config=0;
419     ret=spi_open(&lcd_fd, 0, spi_config);
420     if (ret!=0)
421     {
422         printf("Exiting\n");
423         exit(1);
424     }
425     if (lcd_initialised==0)
426     {
427         lcd_init();
428     }
429     if (dofile)
430     {
431         //linha 1 do arquivo de saída contendo as três colunas de descrição
432         fprintf(outfile, "Time HH:MM:SS,Elapsed Sec,Temp C\n");
433         ...
434     }
435     return(0);
436 }

```

Figura 54 – Código em C, adaptado. Implementado para recuperar valores de conversão de FEM em tensão por meio da interface entre a *Raspberry Pi Zero W* e o ADS1118. Fonte: Yousef, adaptado.

ANEXO C – Terceiro Anexo

A função *isoutlier* foi empregada na identificação de valores discrepantes.

```

1 - filename = 'C:\Users\Filipe\Desktop\TCC2\000 Panilhas de Medidas\Outliers_Matlab_Detection\MedidasParaComparação.xlsx';
2 - %Matriz contendo todas as medidas usando o TM-902C
3 - matrizmedREF = xlsread(filename,'Medidas_TM-902C','B3:F22');
4 - %Matriz contendo apenas as medidas com o sensor ADS1118
5 - matrizmedADS = matrizmedREF(:,1:1:end);
6 - %encontrar os outliers
7 - outlier = isoutlier(matrizmedADS,'mean');
8 - [w,z] = find(outlier);
9
10 %mean = mean(matrizmedADS); %
11 %std = std(matrizmedADS); %
12 %var = var(matrizmedADS);%variancia de cada medida/angulo
13 %coefvar = std./mean;

```

Figura 55 – *Script Matlab* - Uso da função *isoutlier*. Fonte: (MATHWORKS, 2020).